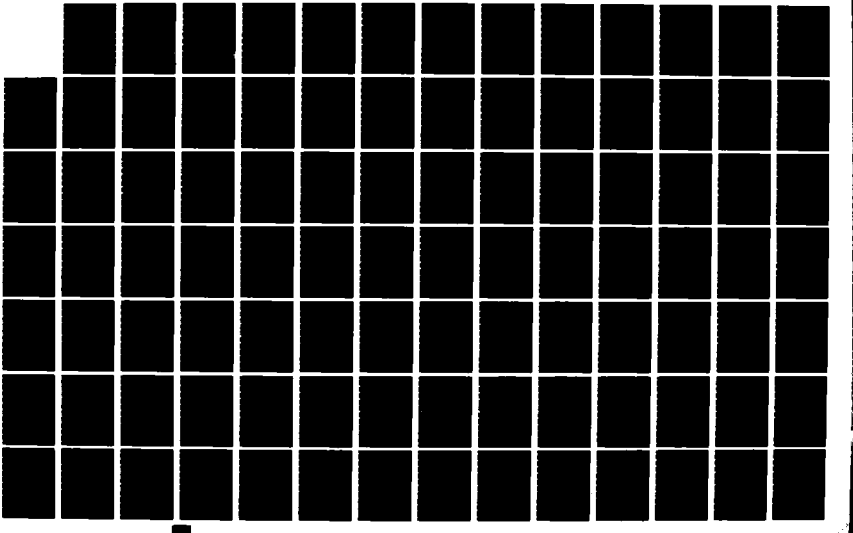
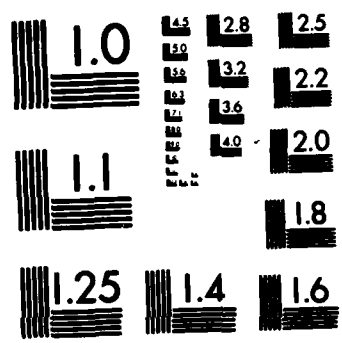
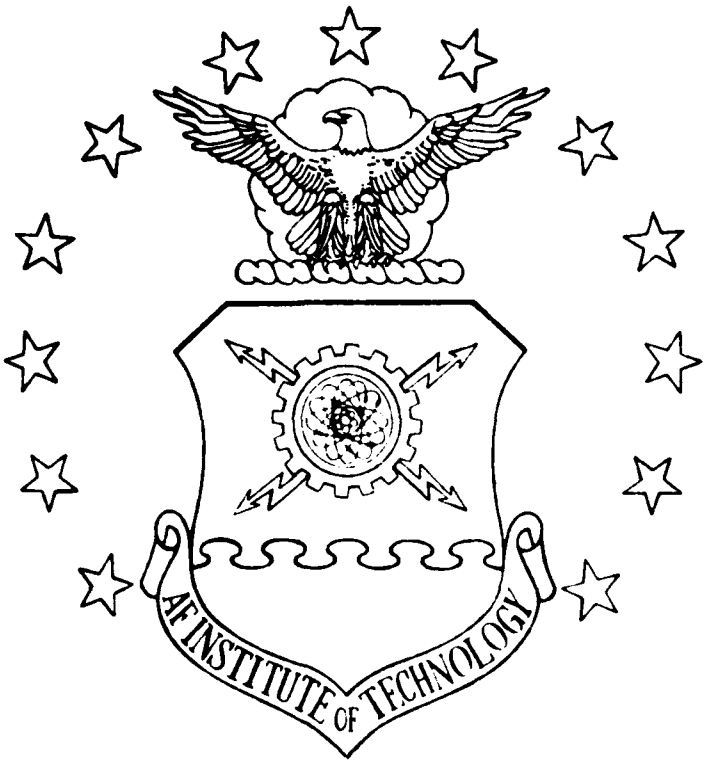
AD-A138 151 A STUDY OF OPTIMAL COMPUTER NETWORK ARCHITECTURE FOR DIGITAL AVIONICS SYSTEMS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. M F KRILIC

1/3

UNCLASSIFIED DEC 83 AFIT/GE/EE/83D-36 F/G 17/2 NL







A STUDY OF OPTIMAL COMPUTER NETWORK ARCHITECTURE FOR DIGITAL AVIONICS SYSTEMS

THESIS

Mensur F. Krilic, Lt Col
Yugoslavian Air Force

AFIT/GE/EE/83D-36

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/EE/83D-36

A STUDY OF OPTIMAL COMPUTER NETWORK ARCHITECTURE FOR DIGITAL AVIONICS SYSTEMS

THESIS

Mensur F. Krilic, Lt Col
Yugoslavian Air Force

AFIT/GE/EE/83D-36

Approved for public release; distribution unlimited

AFIT/GE/EE/83D-36

A STUDY OF OPTIMAL COMPUTER NETWORK ARCHITECTURE FOR DIGITAL AVIONICS SYSTEMS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Mensur F. Krilic, B.S.
Lieutenant Colonel, Yugoslavian Air Force

December 1983

Preface

The purpose of this thesis was to define the characteristics of optimal computer network architecture for digital avionics systems. Previous investigations in the field have indicated that such a definition should be made today to meet the challenging requirements of future avionics.

Although the scope of this work was limited by available unclassified data about military avionics systems, I believe that this report will be found complete and self-sufficient by the reader interested in the subject.

I wish to thank my advisor, Major W.D. Seward who proposed this project and to the thesis committee members LtCol H.W. Carter and Capt J.S. Gordon of the Air Force Institute of Technology. I greatly appreciate their willingness to help and confidence to let me work independently when I needed it.

A special gratitude is extended to Mr. S.J. Larimer of the Flight Dynamics Laboratory, WPAFB for his comments and readiness to review the design of Self-Managing Multiplex System.

And finally, I would wish to thank my wife, Bahra, and our daughters, Amela and Alma, for their generous support and understanding during these very busy months.

Mensur F. Krilic

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AAES - Advanced Aircraft Electrical System
ACU - Avionics Control Unit
AMUX - Avionics Multiplex System
ARPV - Advanced Remotely Piloted Vehicle
BITE - Built-In-Test
BIU - Bus Interface Unit
BCIU - Bus Control Interface Unit
bps - Bits per Second
CRC - Cyclic Redundancy Check
CTIS - Centrally Integrated Control System
DAIS - Digital Avionics Information System
EMI - Electromagnetic Interference
EMUX - Electrical Multiplex System
EOT - End of Transmission
FAA - Federal Aviation Administration
FIFO - First-In-First-Out
FCC - Fire Control Computer
GAMS - General Aircraft Multiplex System
GNACU - General/Navigation Avionics Control Unit
HSMB - High-Speed Multiplex Bus
HXDP - Honeywell Experimental Distributed Processor
ITS - Information Transfer System
LIU - Loop Interface Unit
MC - Mission Computer
MTBF - Mean Time Between Failures
NADC - Naval Air Development Center
NRZ - Non-Return-To-Zero
PE - Processing Element
RT - Remote Terminal
RZ - Return-To-Zero
SIM - State Information Memory
SOT - Start of Transmission
SMS - Self-Managing Multiplex System
TDM - Time Division Multiplex
TSP - Twisted-Shielded Pair
USAF - United States Air Force
WDACU - Weapon Delivery Avionics Control Unit
WDM - Wavelength Division Multiplexing
WPAFB - Wright-Patterson Air Force Base

Abstract

The ramifications of the use of digital avionic systems and the experience drawn from the 10-year development cycle of the MIL-STD-1553B require that solving of the problems due to anticipated growth of avionics should start today. Emerging technologies offer more and more capabilities to avionic systems and optimization of computer architectures is needed to obtain all the benefits that potentially exist.

In this paper, current theoretical considerations in available literature have been used to sort out the essential figures of merit of computer network architectures for digital avionic systems. Fourteen different approaches to the same problem of data multiplexing in avionics systems are analyzed according to the key issues. Conclusions drawn are used to define the "optimal" computer network architecture for digital avionics.

The Self-Managing Multiplex System (SMS) is conceptually designed with respect to the "optimal" characteristics, along with the discussions of some trade-offs that had to be made. The burst errors self-correcting feature of the broadcast-acknowledgments in the SMS concept seems to deserve some sort of testing in practice. It is recommended that a detailed simulation study should be performed later and a hot bench built up using the latest technologies that exist.

A STUDY OF OPTIMAL COMPUTER NETWORK ARCHITECTURE FOR DIGITAL AVIONIC SYSTEMS

I. Introduction

At a first glance, the transmission of multiplexed data over a pair of simple twisted-shielded wires might be considered as a problem that is not broad enough to warrant the thesis leading to the Master of Science Degree in Electrical Engineering. However, the subject is far reaching indeed, simply due to the fact that the pair of wires represent the internal time division multiplex data bus -- the "artery" of any digital avionic system. All impacts of data multiplexing can be understood only when the ramifications of the use of digital avionic systems are considered: since the first digital computer as has been employed in the F-106 in the late 1950's, up to the so-called "all-digital aircraft" as foreseen for the 1990's.

Digital avionics multiplex techniques have a number of definite advantages over their analog counterparts. Elimination of unnecessary replication of information sensing and display, performance and reliability gains, reduction of the weight, size and costs, and lack of space are usually given as the major reasons for the digital avionics usage and integration (Ref 3:2-2). As pointed out in the same reference, digital avionics integration is the cooperative use of

shared information among avionic subsystems, which became necessary when requirements for missions and their associated avionic hardware could no longer be met practically in an aircraft with independent and self-sufficient subsystems.

On the other hand, the classical avionic systems design methodology was to provide unique systems for the specific aircraft. Different manufacturers have produced the hardware with little or no relationship to each other. The proliferation of "black-boxes" in the aircraft that were part of the same fleet, but without any compatibility or commonality of either hardware or software for their avionics, resulted in the excessive overall avionics design, acquisition, and maintenance costs that tended to restrict the number of aircraft procured. In that sense, not only integration within a single aircraft was needed, but the integration had to be done in a way that provides standardization among the avionic systems even when they were intended for completely different missions. The time division multiplex data bus is, undoubtedly, the cornerstone of both the digital avionics integration and standardization.

The standardization efforts started in the late 1960's-early 1970's and coincide with the F-15's, B-1's, and F-16's development or early procurement stages. As the first result, the USAF issued its MIL-STD-1553 standard in August 1973. Based on the application requirements for the USAF, US Navy,

and US Army, the standard has been upgraded and published as the tri-service document, the MIL-STD-1553A on 30 April 1975. The current update, MIL-STD-1553B was issued on 21 September 1978 with the Notice 1 (USAF) from 21 February 1980. The standard is accepted by Great Britain, as well.

The technology is considered mature today. The MIL-STD-1553A/B-based hardware components, software solutions and/or entire system architectures can be found almost everywhere throughout the aerospace industry. More and more weapon systems are built around the MIL-STD-1553 bus today. According to the latest published data (Ref 31), this is the case even when severe environmental, space, weight, and power restrictions limited the amount of computing capability available to support bus operation. An example is the MSRAM (medium range air-to-surface missile) which is an adaptation of the Tomahawk cruise missile.

Digital avionics technology continues to increase in number of applications and complexity by an order of magnitude every few years (Ref 25:48). Today, avionics account for about one-third of the aircraft's total cost. In parallel, the MIL-STD-1553 philosophy has proven itself to be an effective way for avionics integration and will, very probably, stay that way well into the 1990's. The question then arises: 'What is the purpose of further research to change an effective standard that fits so well to such complex and

expensive equipment?'

The twofold answer to that question covers the scope of this thesis, as well:

1. The anticipated growth of avionics will require more and more processing capabilities and, therefore, more and more data to be transferred at high rates. Having in mind the 10-years development cycle of the MIL-STD-1553B, it is obvious that solving problems of tomorrow should start today. In addition, some inherent inflexibility of the MIL-STD-1553 technique, which is hidden by the undisputable advantages of the approach, might cause a lag in the ability to exploit all benefits from the emerging technologies in the future.

2. The exact network topology, data protocol, and configuration have not been standardized for the avionic subsystems themselves, although they must be designed around the MIL-STD-1553 data bus. An analysis is required to determine the best subsystem network architecture for optimal system response and reliability.

In fact, this is a subject which has received much attention in recent years. Nearly every organization involved in aerospace system design has its own research program in this area (Ref 53:309). A huge amount of literature exists, with sometimes diametrically opposite opinions expressed. The survey of literature search, as given in Chapter II of this thesis, will help the interested reader

locate valuable information to start his own research. In addition, throughout the report numerous citations from a variety of references are made to provide the reader more insight into the subject area.

The discussion in Chapter III is based on existing general theoretical considerations about the figures of merit of different philosophies. The key issues are sorted out in an attempt to help description and analysis of a variety of approaches to the same problem of data multiplexing.

In Chapter IV, fourteen different techniques are described and analyzed with respect to the key issues discussed in Chapter III.

The conclusions drawn from both Chapter III and Chapter IV led to a definition of what the "optimal" architecture should be. The characteristics of the architecture are listed in Chapter V. In order to resolve some contradictions in this set of characteristics of an "optimal" architecture, and to give guidance how to synthesize a system by resolving the contradictions on a problem-by-problem basis, the Self-Managing System (SMS) has been introduced in the same chapter. Except for the concept that allows acknowledgments in broadcast mode, the technique is not a completely new idea. Rather, it is an integration of several different techniques in a new context.

The report concludes in Chapter VI. A limit on the study was the security clearance required to obtain complete information about existing solutions on military aircraft, including their weaknesses and strength. The author had no such a permission and the results are based on the available data from unclassified literature. This means that the assumptions made might not be true in filed applications. To truly prove a suitability of the SMS technique, a recommendation is made in Chapter VI that a detailed simulation study should be performed later and a hot bench built up using the latest existing technologies. At least, the burst errors self-correcting feature of the broadcast-acknowledgments concept seems to deserve some sort of testing in practice.

II. Results of Literature Search

In order to identify and evaluate sources of information, two separate literature searches were conducted: through the National Aeronautics and Space Administration (NASA Literature Search Number N49276 / May 1983), and the Defense Technical Information Center (DTIC / April 1983). In addition to that, the author has performed an independent search over available literature at the Academic Libraries at the AFIT School of Engineering and School of Systems and Logistics, the AFWAL/TST Technical Library in WPAFB, and the Wright State University Library in Fairborn, Ohio.

These searches include accessible unclassified journals, reports, and technical information published since 1968 on both the existing and proposed solutions pertaining to digital avionics integration, multiplexing, and architectures. In that way, the period from early days of digital avionics and the first attempts for standardization in the field is covered.

More than five hundred references were reported in these searches, showing continued efforts made over the past 15 years to find an optimal solution in the area. The highlights of the literature, as a combined list of references and bibliography, can be found at the end of this report.

In spite of the fact that there are many apparent

controversies involving both newly proposed and past modeling techniques, it is noticeable that avionic architectures are generally evolving towards more sophisticated capabilities offered by hierarchical, distributed control architectures.

On the other hand, the main characteristic of present solutions is an architecture based on the MIL-STD-1553 Aircraft Internal Time Division Command/Response Multiplex Data Bus (Ref 65). Although there is not a reference without some sort of description, critique, or eulogy of this solution, the most complete survey of the bus can be found in the Multiplex Application Handbook (Ref 3), with shorter versions offered by Ref 21 and 48. Useful literature about this topic is also represented by Ref 36, 37, 62, 73, and 75, while a detailed analysis of some of the main physical characteristics of the bus is given in Ref 57. A comparison based on a computer simulation of three somewhat different implementation of the MIL-STD-1553 based data buses is offered in Ref 58. Many solutions can be found in literature for bus controllers, remote terminals, different types of testers, and simulation design tools pertaining to the MIL-STD-1553 architecture (Ref 5, 14, 17, 29, 33, 44, 47, 60, 64, 78, 84, and 91).

Having in mind that incompatibilities exist between military and commercial aircraft data multiplexing systems,

SCI Systems Inc. developed the Programmable Interface Module (Ref 41) which could be used in achieving the compatibility between different buses.

Air superiority, close air support, manned bomber, and remotely piloted vehicles are recognized in Ref 20 as having a large core of common equipment types performing fairly similar functions across the various aircraft. This fact led to the definition of the general system framework known as the Digital Avionics Information System (DAIS), being under development by the Air Force since 1973. DAIS provides a system architecture which can adopt currently available avionics subsystems (radars, inertial platforms, etc.) to a common, modularly expandable avionics core (Ref 87), representing the second most often referenced concept throughout the literature (Ref 8, 9, 10, 11, 12, 16, 23, 32, 59).

Results of an investigation of functional commonality of avionics systems specific to Naval aircraft, with general requirements for the development of core Naval aircraft avionics equipment are given in Ref 22.

An attempt for computer system optimization for avionics application is described in Ref 15. This was made by an examination of the commonalities among the four subareas of signal processing (radar, communications, image processing and electronic warfare), in order to determine computational complexity and commonality among kernels and algorithms for

architectural purposes.

Conceptual design considerations for an integrated power and avionics information system can be found in Ref 18, 28, and 38.

Almost immediately after the adoption of the first version of the MIL-STD-1553 architecture, some disadvantages of the concept have been recognized (Ref 80) and the searches for other solutions are still continuing (Ref 19). The main goal is to achieve an architectural concept that will allow multimission applications along with increased data rates and improved reliability, survivability, availability and affordability. This has been analyzed and presented in an excellent study made by the Boeing Military Airplane Company (Ref 81) in late 1982.

The McDonnell Aircraft Company is currently performing further studies of an advanced avionics system specification within the Pave Pilar Advanced System Integration Demonstration (ASID) Definition Program. This study is contracted (F33615-82-C-1904) by the Avionics Laboratory Division of the Air Force Wright Aeronautical Laboratories (AFWAL) and will be done in 1984; the initial results were presented at the NAECON '83 (Ref 76).

In fact, nearly every military or civilian institution dealing with design of airborne systems has its own research program in this field (and its own opinion and approach to

the problem!), as follows:

In the Avionic Multiplex System (AMUX) developed by the Vought Corporation for the Advanced Aircraft Electrical System (AAES) Simulator at the Naval Air Development Center (NADC), both the command/response and polled contention modes are implemented (Ref 90), capitalizing in this way upon the dynamic bus allocation feature of the MIL-STD-1553 concept.

But, this is not an isolated attempt to fully explore all of the existing features of present standard. In Ref 4, the British Aerospace PLC addresses problems with regard to the handling of acyclic message transfers either on the same bus or between buses, and to the broadcast transfer checking schemes. The High Speed Multiplex Bus (HSMB) Protocol Study, made by NADC (Ref 86), makes not only the use of MIL-STD-1553B functions as a subset of an HSMB protocol, but also goes beyond to completely specify the broadcast mode of operation and a more detailed implementation method for dynamic bus control allocation. A so-called Hybrid System is recommended by Texas Instruments Inc. (Ref 2) in an attempt to design an avionic digital processing system for multimission Advanced Remotely Piloted Vehicle (ARPV) applications.

A bus-oriented modular avionic architecture with distributed processing seems to offer highly desirable attributes of growth and independence from technology implementa-

tion details. This is the philosophy advocated by General Electric in Ref 67, using the US Navy's Advanced Integrated Display System as the example. It must be noted that this philosophy is, in a way, the ultimate goal being implicitly or explicitly advocated by almost everybody. Controversies arise concerning different opinions as to what is the "best" way to achieve that goal.

In Ref 63, Boeing Military Airplane Company pleads for contention multiplex systems and therefore for the subsystem independence that was lost by integrating with a central minicomputer. The quantitative performance of three various types of contention Information Transfer Systems (ITS) is presented and the conditions for which particular ITS would be appropriate are discussed.

Ref 53, 55, and 61 are showing the efforts made by the AFWAL to achieve an architecture for event-driven real-time systems, which is diametrically opposite to the MIL-STD-1553 concept and represents an attempt to explore all potential benefits of distributed architecture and purely (so-called "transparent") contention protocol.

General Dynamics proposed a switched network architecture (Ref 50) that works very much like the public telephone system, dropping out the bus oriented technique, but still providing a support for modular structures.

In an excellent tutorial of avionics systems architec-

tures (Ref 49), the Honeywell Systems and Research Center explores the implications of fiber optic interconnections and intelligent (i.e., microprocessor-based) communication interface units on the architecture of distributed processing systems. As a conclusion, a call is made for continuous and considerable research and development efforts in the area of decentralized executive control before all the advantages can be realized.

A concept presented by the Forecasting Integrated Technologies Corporation (Ref 45) concludes this review of newly proposed solutions: an advanced memory module is proposed as a medium to interface between the aircraft crew and its operational systems in such a manner as to synergistically integrate crew and aircraft into a more effective weapon system. The paper does not argue with the fact that the distribution of processing resources leads to significant avionics performance increases, but advocates that the proper application of mass memory can add more to promote avionics performance.

Of course, there is not a single discussion in the literature which does not address fault tolerance aspects in the field, but good insight into the problem can be obtained from Ref 19, 43, and 77.

Very useful general knowledge literature is given in Ref 7, 13, 24, 30, 34, 35, 40, 51, 70, 71, 83, and 92.

III. Characteristics of Avionics' Architectures

The fact that avionics data multiplexing has accumulated so much attention during recent years makes any analysis and documentation of current knowledge in the field a cumbersome task, especially if the goal is to synthesize an architecture based upon conclusions drawn on the "advantages" and/or "disadvantages" of existing technologies. However, if that is to be done, then three sets of important issues should be defined:

- First, one has to characterize digital avionics network architectures with enough generality to allow a description and documentation of different (and sometimes mutually exclusive) approaches in a similar manner;

- The second set is an evaluation tool to compare the attributes of presented techniques with respect to the impact of implemented philosophy on overall system performances; and finally,

- The synthesis of the "best" architecture should be performed according to the third set of selected criteria, recognizing all possible conflicts and necessary tradeoffs, then making decisions on a problem-by-problem basis.

It is obvious that the issues arranged in such or similar manner are never mutually independent and, probably, this is the reason that the analyses performed so far are

generally made based on one common or at most two (but interlaced) sets of criteria (Ref 19, 49, 80, etc.). It seems, in fact, that an attempt to develop uniquely defined and universal criterion to judge all possible solutions would have no useful application in the real life -- the "optimum" is too much dependent upon many contradictory factors, including the role and the size of the aircraft. This has been born in mind throughout this study, leading to the final conclusion that the "optimal" architectural approach would be the one that captures as much as possible of the existing techniques, and that is reliable, simple, and general enough to be standardized for use in the entire aerospace industry, but -- ABOVE ALL -- that leaves system designers free and technologically independent to implement the philosophy according to the particular information system requirement, allowing at the same time a possibility for subsequent growth of the system with relative ease.

However, the author felt that for the purpose of this study an arrangement of the key issues should be made somehow, to help the organization of the study. Thus, the multiplex system "descriptive" and "analysis" issues as the sets of criteria have been chosen, as discussed in next two sections. The "synthesis set" is implicitly defined in Chapter V along with the development of the proposed architecture.

Multiplex System "Descriptive Issues"

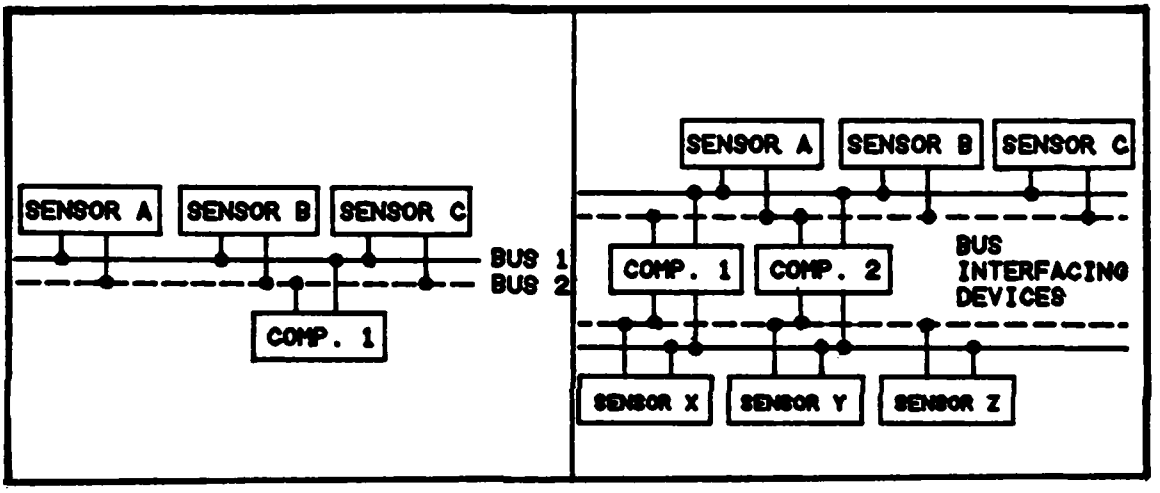
Classification of systems by this set of issues should allow the system characteristics to be analyzed within some combination of generic subcategories. These subcategories are represented mainly by the system's general characteristics (intended mission, topology, architecture, control method, and message protocol) and by the physical/electrical characteristics of the multiplex bus (transmission medium and coupling method, maximum bus distance, signal modulation, and transfer synchronization method). These issues seem to be sufficient (Ref 19, 49, and 80) for the purpose of system description and categorization, as follows:

System Architecture. The term "architecture" refers to the way subsystems are connected together, including the method to control data transfer.

Bus Control. The bus control method or bus allocation scheme (as defined in Ref 49:7) is the mechanism which ensures that exactly one terminal at a time has the right of the multiplex bus usage. This issue is discussed in the Multiplex System "Analysis Issues" section.

System Topology. The topology of a data bus system is the map of physical connections of each subsystem to the data bus. There exist two general types of data bus topologies (Ref 48:I-25): single level and multiple level (hierarchical). The single level bus topology is the simplest

scheme and is the most commonly used interconnect architecture today. In this approach each of terminals is connected to a single level bus system (Figure 1a) regardless of the data bus redundancy -- the use of multiple buses for redundancy in a single level system does not change the type of topology.



a) Single Level Bus Topology b) Multiple Level Bus Topology

Figure 1. Bus Topologies (Ref 48:I-26)

Multiple level or hierarchical bus topology is an extension of the single level concept. If single level buses are interconnected in a certain manner, data on one bus system will pass to another bus system (Figure 1b). Such interconnection scheme can be used to obtain functional partitioning in a system. This is, also, an important issue and will be discussed in more detail later on.

The reader should be aware, that with the increase in the total number of processors added to the automated environment of modern aircraft, the choices of processor interconnection topology and methodology becomes more and more critical. The choices adopted will profoundly influence the information throughput, the fault tolerance, the reliability and maintainability, and the all pervading integrity of the entire avionics system (Ref 44:662).

Message Protocol. The message protocol is referred to as the message exchange sequences provided, as well as to the message formats themselves (Ref 41:12).

Transmission Medium. The transmission medium selected determines to a large degree the error rate and power requirements of the multiplex system. Some types of media that have been considered (Ref 40:230-233) are fluid and acoustical waves, electro-magnetic radiation, optical links (fiber-optics and free space), and several types of wire cables (coaxial, triaxial, twin axial, twisted-shielded pair, planar parallel flexible, and twin-lead).

Fluid and acoustical links are possible and may be considered for special applications. They are impractical for use as a primary transmission medium, however, due to long propagation delays and sensitivity to vibration.

Electromagnetic radiation guided by means of dielectric waveguides has received considerable attention due to

the high bandwidth and attendant high data rates possible with this method. Waveguides, however, are still bulky, expensive and highly susceptible to physical damage.

Optical data transmission links continue to show great promise by virtue of the extremely wide bandwidth obtainable. Bandwidths on the order of 100 GHz can be realized using fiber optics. Coupling, termination, and repair of fiber optics remains a problem at present, however.

For these reasons, wire cable has received primary consideration as a multiplex transmission medium. Of all types considered, coax, triax, twinax and twisted-shielded pair cables have shown the greatest potential. While coax and triax are more suited to high frequency applications (up to 100 MHz), electromagnetic interference (EMI) susceptability and radiated noise are higher due to the fact that the shield carries signal current. Twinax and twisted-shielded pair (TSP), on the other hand, are usable at frequencies of 10 MHz and are balanced configurations in which the signal carrying conductors are shielded. It is apparent that twinax or TSP cable is more desirable at the relatively low data rates associated with most aircraft applications. Although twinax is clearly superior to TSP in its bandwidth and attenuation characteristics, TSP is more attractive in many applications by virtue of its extremely low cost. On relatively short bus lengths, such as would be encountered on a

light-weight fighter type aircraft, the slightly higher attenuation of TSP is not significant.

Coupling Method. Coupling consideration is another important issue in the physical/electrical characteristics set and the same reference as cited above (40:230-233) is probably the shortest but still complete discussion that can be found about the matter.

Several methods are available for coupling the users to the multiplex transmission link:

- Direct coupling, which is a simple and inexpensive method, but offers no isolation and presents a matching problem;

- Capacitor coupling, that provides DC isolation, but must be bandpass filtered at higher frequencies to prevent ground loops;

- Transformer coupling, that maintains isolation, simplifies matching, and is relatively nonlossy, making it the primary coupling method today. One of the disadvantages is that the transmitted signal must have no DC component, but this problem can be easily overcome by judicious selection of the modulation method.

The coupling problems affect the system topology (the transmission link configuration - Ref 40:231) and could well be one of the most important considerations in the design of a multiplex system, especially where the transmission link

will be subjected to a high-risk environment. Two basic configurations possible are the lossless and lossy configurations. The lossless line is a classical approach to the design of a transmission link. Its basic configuration is as shown in Figure 2. By terminating the line in its characteristic impedance (Z_o) and properly controlling the terminal impedances, reflections are kept to a minimum. In the receive mode, the terminals are held at a relatively high impedance thereby minimizing their effects on the line and in the transmit mode are brought to a low impedance.

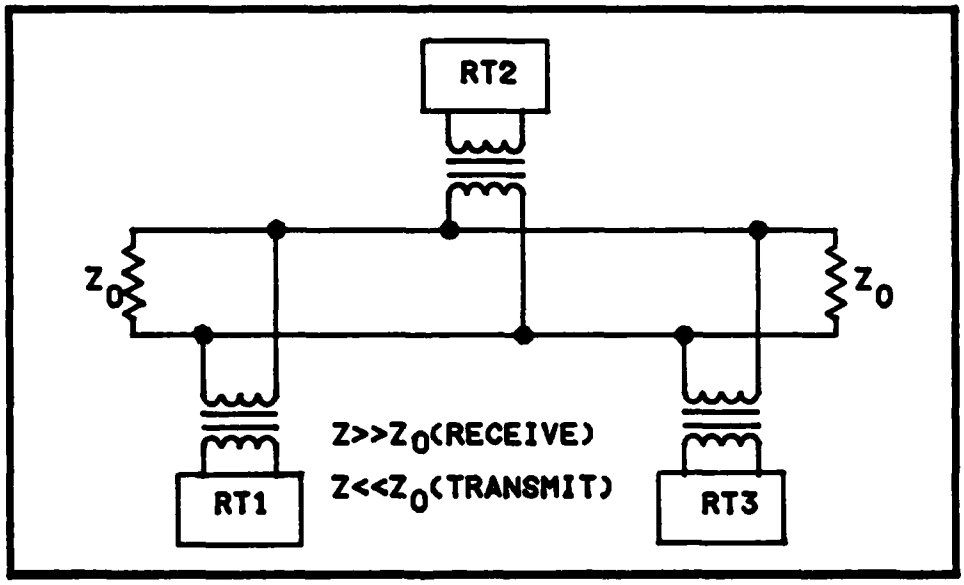


Figure 2. "Lossless" Line Configuration (Ref 40:231)

The line is "lossless" only in that an attempt is made to keep losses at a minimum and no intentional losses are introduced. Since losses are relatively low, however, the lossless line does require less transmitter power to achieve

a given S/N ratio in a worst case condition. From a purely analytical viewpoint, the lossless line is superior to the lossy line in weight, cabling required, minimization of reflections, hardware simplicity, and the ability to easily increase the length of the line.

The major disadvantage incurred through the use of the lossless line is its inability to withstand line faults. As can be seen in Figure 2, a line-to-line short at any point places a direct short across the transmitting terminal and the entire system is lost. Should the line open, a severe mismatch would occur and perhaps system would be lost, too.

The only means of protection against line faults with the lossless line is to provide multiply-redundant lines to whatever level is required to ensure an adequate probability of mission success. Redundant lossless lines, however, require active switching elements to select the appropriate line, thereby adding hardware complexity.

The basic lossy line configuration is shown in Figure 3.

By providing loss at each terminal and looping the line, a configuration is formed which is significantly more immune to line faults than the lossless approach. Configured in this manner, the line can tolerate one fault, either open or short anywhere on the link without loss of any of the terminals. If two lossy lines are connected in a dual-crosscoupled configuration as shown in Figure 4, both lines can withstand several

faults without the loss of the multiplex system.

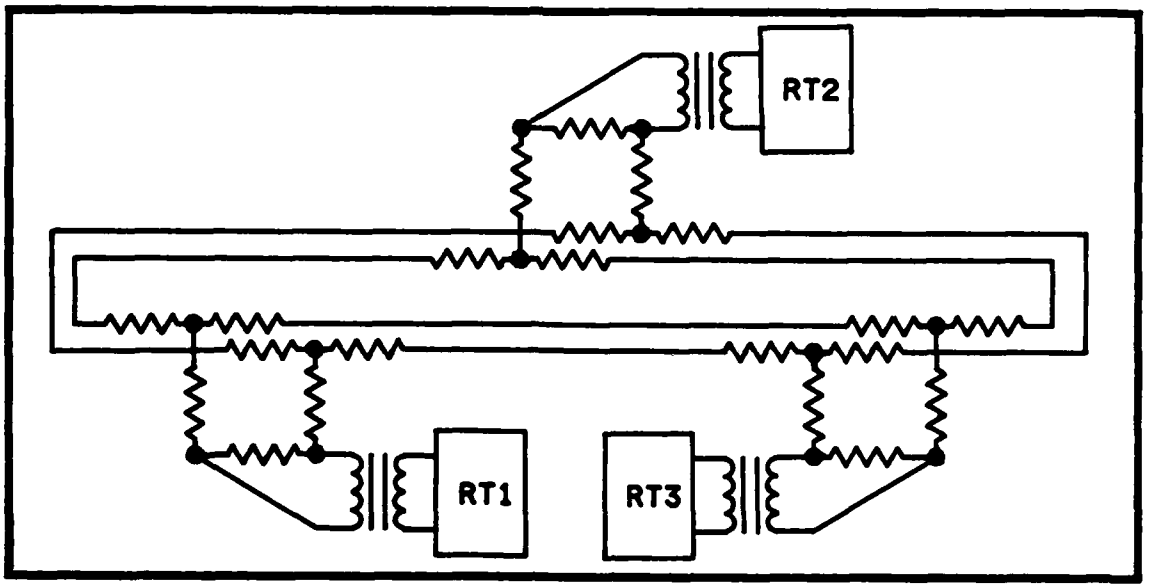


Figure 3. Looped Lossy Line Configuration (Ref 40:232)

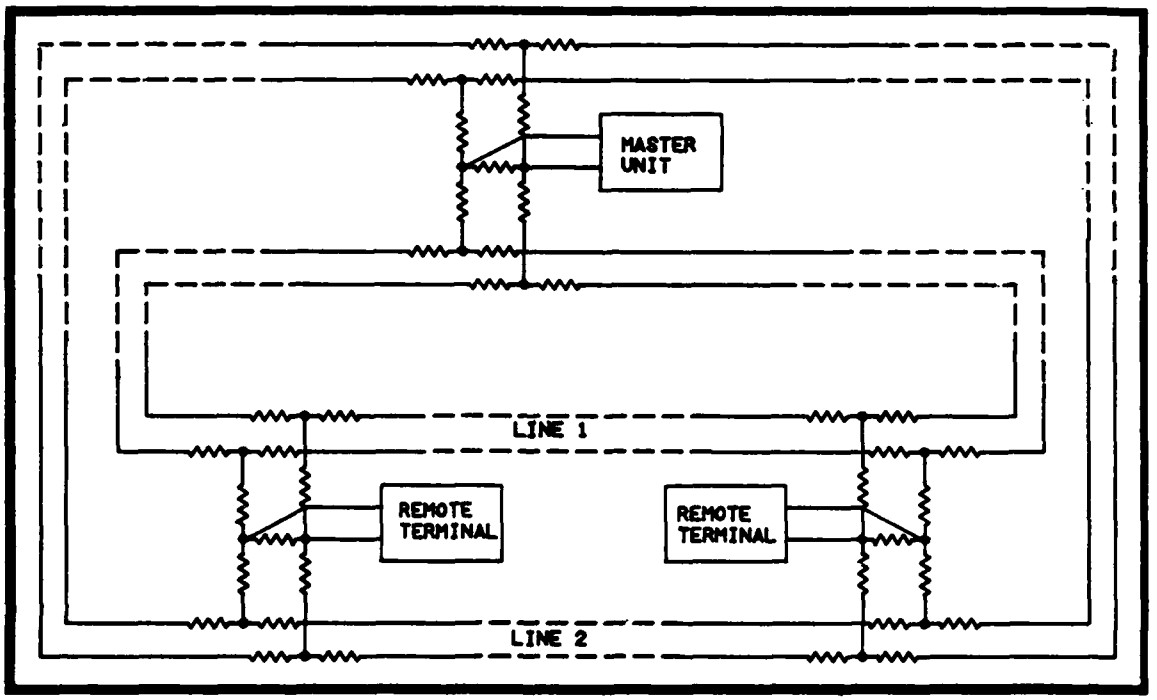


Figure 4. Looped Lossy Cross-Coupled Configuration (Ref 40:232)

Additionally, the lossy line provides a form of "graceful degradation" in that individual terminals may be lost one at a time.

For the dual redundant lossless configuration it can be shown (Ref 40:232) that the probability of loss of one or more terminals due to line faults is

$$[1 - (1-P)^{N-1}]^2$$

while for the looped lossy cross-coupled configuration, the probability of loss is

$$\sum_{i=2}^{N} \binom{N}{i} (P^2)^i (1-P^2)^{N-i}$$

where "P" is the probability of one or more faults occurring on a line segment between two adjacent terminals and "N" is the number of terminals in the configuration. By assuming seven terminals, a graph can be plotted comparing these two configurations for different fault probabilities as shown in Figure 5.

It can be seen that the looped lossy cross-coupled configuration is considerably less susceptible to line faults. This is an important consideration in a high-risk environment where the probability of battle damage is great.

Several disadvantages are associated with the lossy line approach. Among these are increased weight, additional cable required, and increased complexity of the line. Additionally,

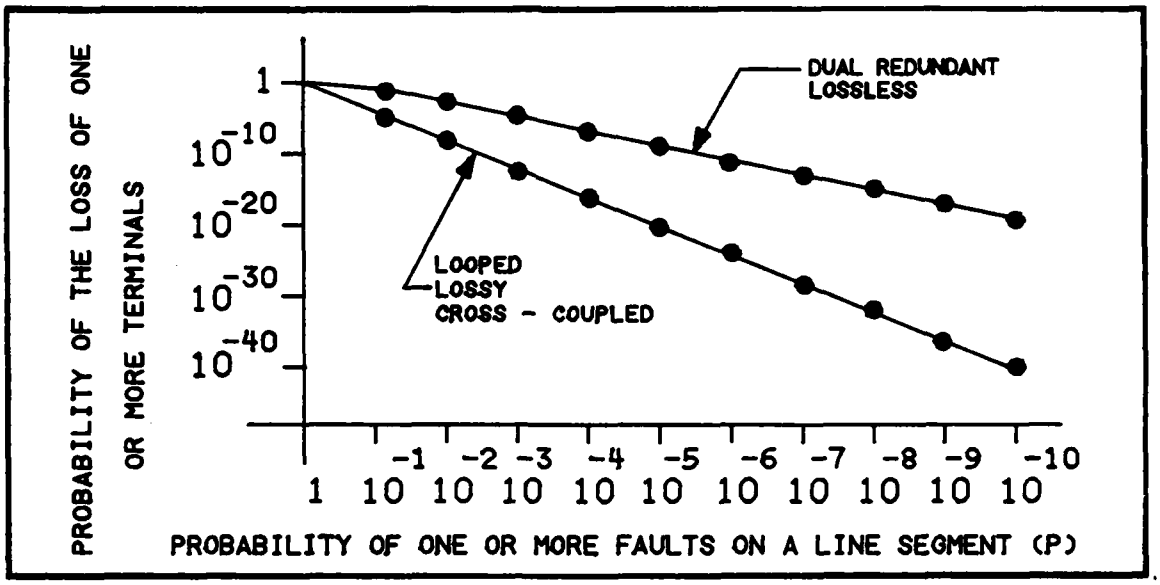


Figure 5. Comparison of Lossless and Lossy Configurations (Ref 40:232)

the receivers must be designed to operate over a wider dynamic range and more signal power may be required than with the lossless line. A disadvantage associated with looping the line is the fact that propagation delays may result in the signal arriving at different times at a given terminal due to the different lengths of the two paths around the loop. This limits the maximum length of the line depending on the data rate involved.

Signal Modulation. As already stated, the coupling method and signal modulation are not mutually independent, especially if baseband techniques are to be used to exploit the advantage of the technique that does not involve a carrier. In the Figures 6 through 11, the minimum transmis-

sion bandwidth (B) is expressed in terms of the signaling rate (f_s) in bps, where the signaling rate is the highest frequency that must be transmitted to ensure that the message can theoretically be received. The $S(\omega)$ represents power spectrum.

Of some 23 different baseband modulation techniques possible (Ref 40:233), the seven are the main candidates for airborne use as discussed in the literature (Ref 3:2/21-2/25):

Unipolar NRZ modulation (a plus level represents a "1"; zero level represent "0") is the inherent form in which data are customarily expressed in logic circuits (Figure 6). Because it is incompatible with transformer coupling, it is not as suitable as some other methods for use on the primary data bus. In addition, it requires twice as much signal power for a given peak-to-peak signal excursion as polar NRZ, which otherwise has identical properties.

Polar NRZ (same as unipolar NRZ level, except waveform is symmetrical about zero) is easily generated from unipolar NRZ by subtracting out its DC level (Figure 6). For this reason, it is less suited for internal logic operations than unipolar NRZ, but is better suited for applications on short direct-coupled local buses because it requires only half as much signal power for the same peak-to-peak signal excursion.

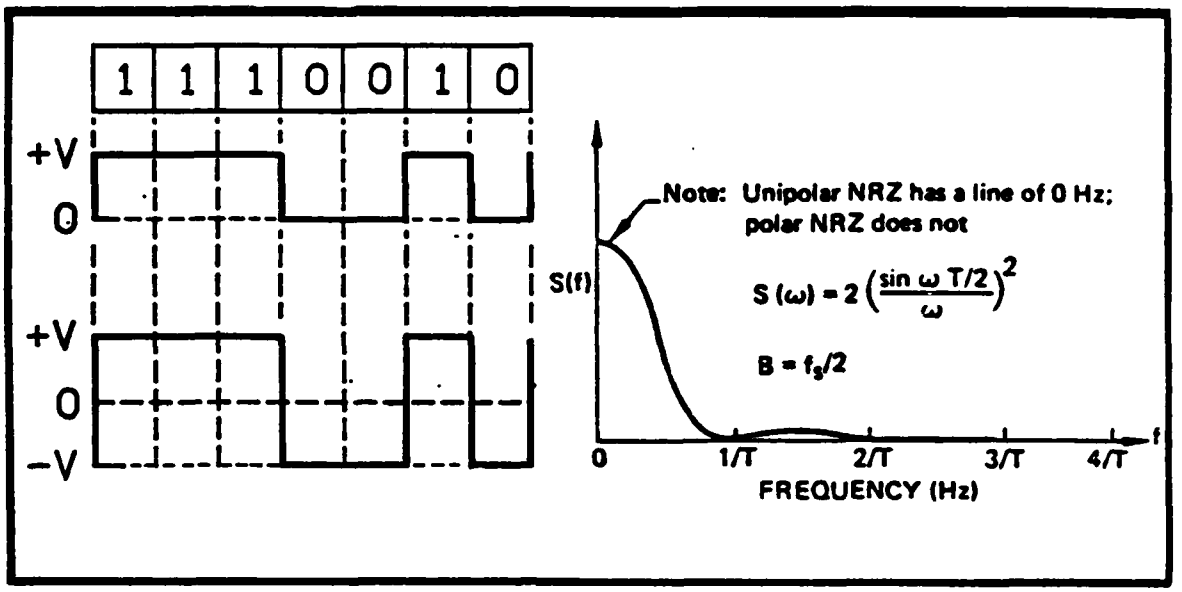


Figure 6. Unipolar and Polar NRZ Modulations (modified from Ref 3:2/22)

Polar RZ modulation is depicted in Figure 7: a "1" is represented by +V pulse and "0" is represented by -V pulse; the pulses are 1/2 bit wide.

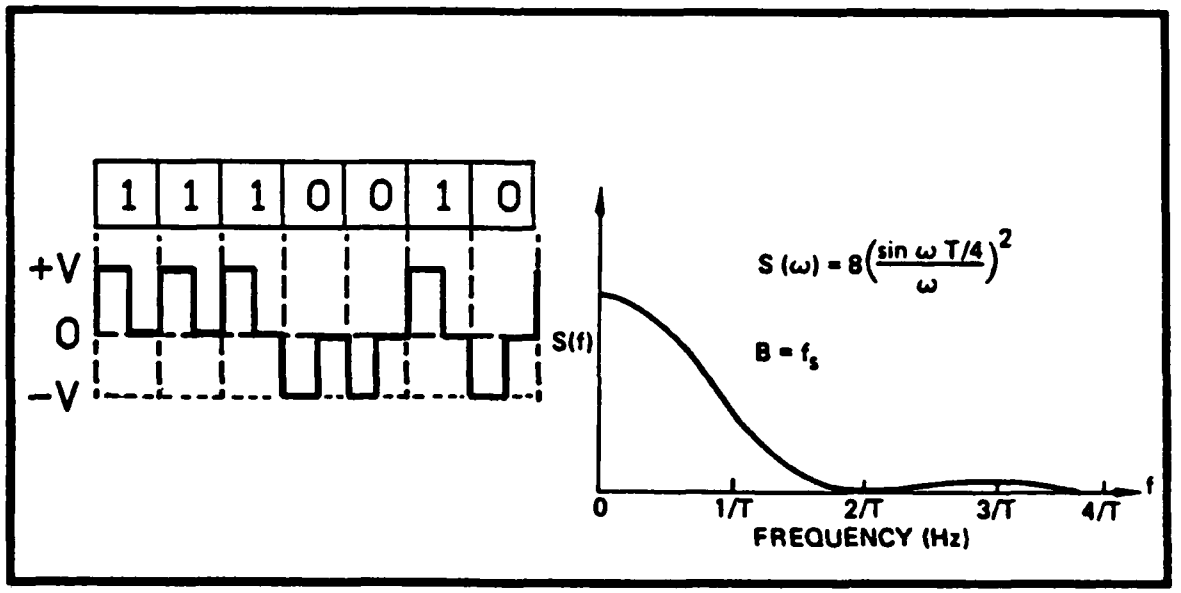


Figure 7. Polar RZ Modulation (modified from Ref 3:2/22)

Bipolar NRZ (Figure 8), because of its greater complexity is less appropriate than polar NRZ on short local direct coupled buses and, also, does not contain bit synchronization information in the basic signal waveform ("1's" are alternately +V or -V pulses; 1-bit-wide zero-level represents "0").

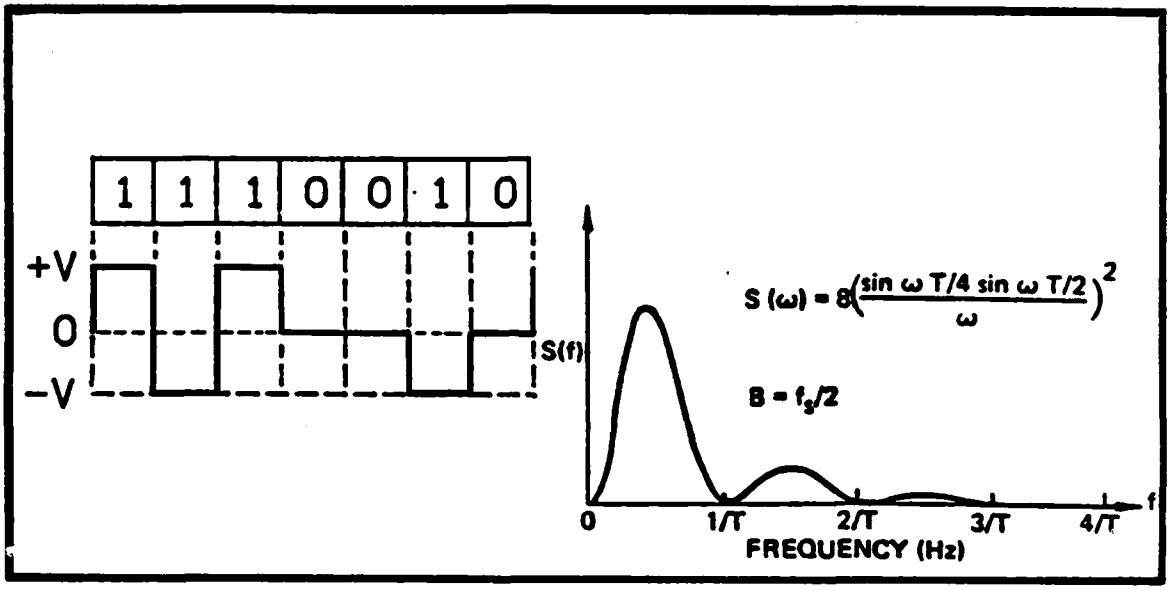


Figure 8. Bipolar NRZ Modulation
(modified from Ref 3:2/22)

In the delay modulation method (Figure 9), which is also known as the Miller code, a "1" causes transition in middle of bit interval; "0" has no transition, except at the end of bit interval between adjacent "0's".

Duobinary modulation is depicted in Figure 10: "1" is zero level; "0" is either +V or -V, the polarity being reversed each time a "0" is preceded by an odd number of

consecutive "1's".

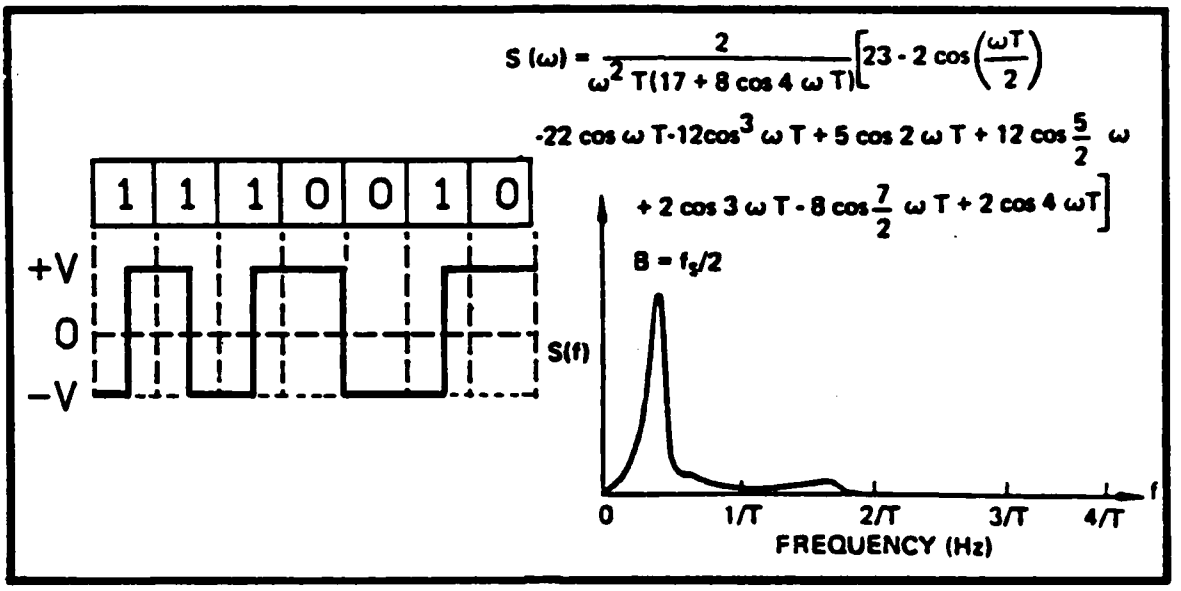


Figure 9. Delay Modulation (modified from Ref 3:2/23)

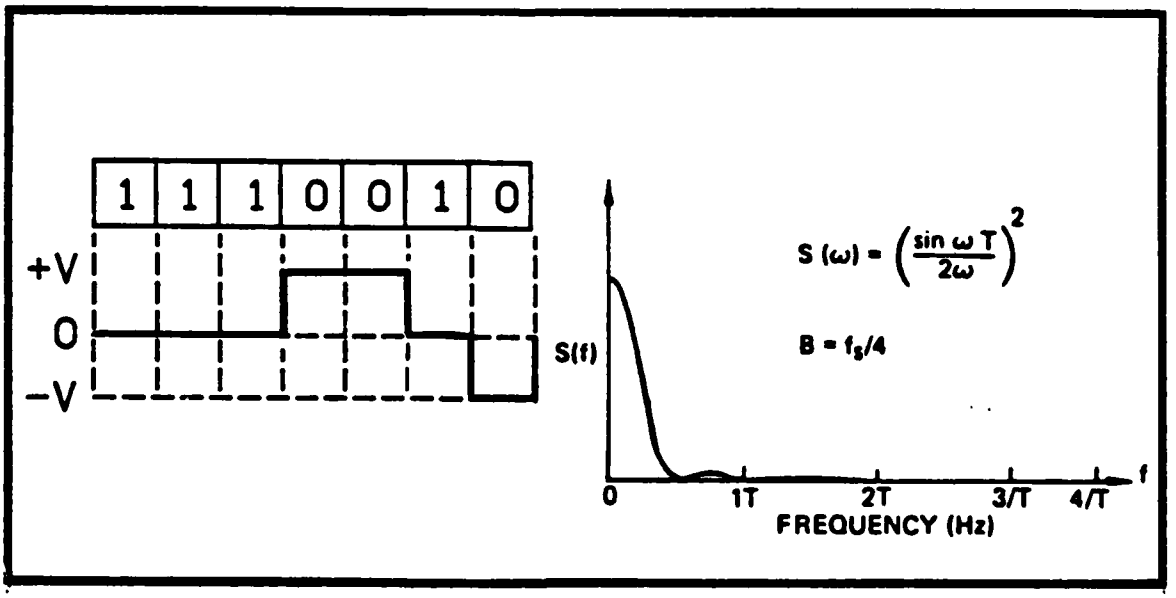


Figure 10. Duobinary Modulation
(modified from Ref 3:2/23)

The BiPhase-Level modulation technique (Figure 11) is also known as the Manchester-II code and can be found in most airborne multiplex applications. In this technique, "1" causes transition from +V to -V level in the middle of a bit, and "0" transition is from -V to +V level, also in the middle of the bit.

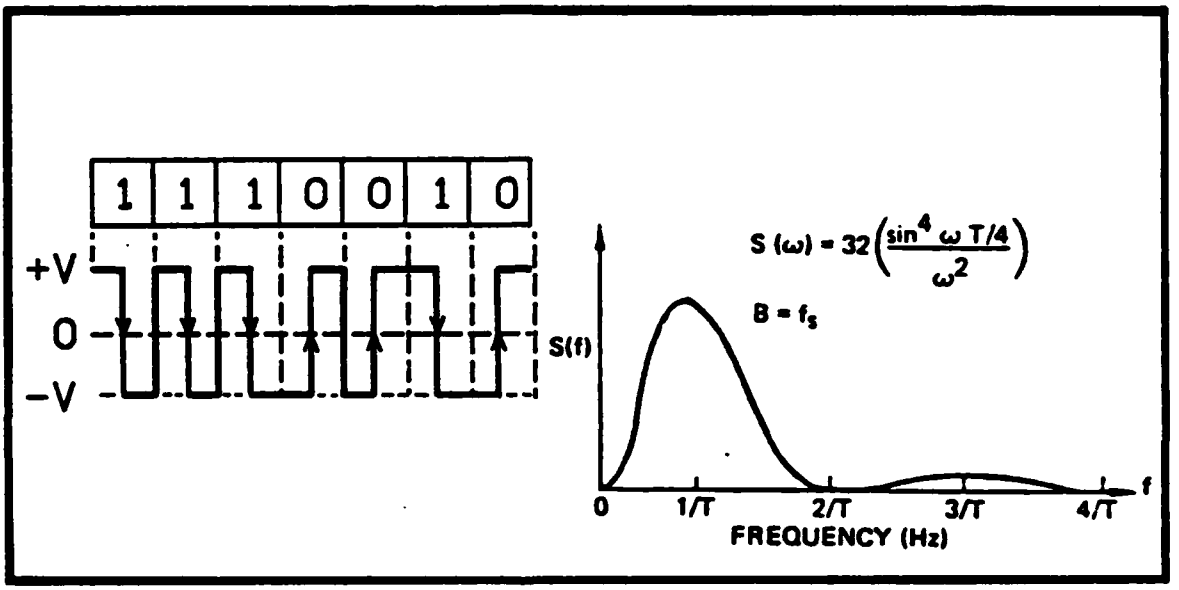


Figure 11. BiPhase-Level Modulation
(modified from Ref 3:2/23)

A comparison made between these modulation techniques can be used to summarize their advantages/disadvantages:

Neither unipolar NRZ nor polar NRZ is compatible with transformer coupling. They, also, do not contain adequate bit sync information, and are useful on direct-coupled short local buses only in those applications where bit sync is conveyed by some other means (such as via a separate channel).

In this case, the polar NRZ is superior to polar RZ because the polar NRZ is less complex and requires only half the transmission bandwidth. The polar NRZ is also superior to duobinary modulation method, because it is far simpler and because bandwidth conservation is not a serious problem when the cable length is short.

Polar RZ is superior to duobinary and unipolar or polar NRZ in direct-coupled local bus applications, which require that bit synchronization information be contained in the basic signaling waveform.

Bipolar NRZ, as stated, because of its greater complexity is less appropriate than polar NRZ on short local direct coupled buses and, also, does not contain but synchronization information in the basic signal waveform.

Bipolar NRZ may be more appropriate than BiPhase-Level in cases where a separate clock channel is available and bandwidth is a significant factor.

The delay modulation method is less appropriate than polar or bipolar NRZ on short direct-coupled local buses that do not require that clock information be extracted from the signaling waveform. If such requirement exists, then the polar RZ is more suitable.

Duobinary modulation is not appropriate for any applications that require transformer coupling, and can find the usage when bandwidth restriction prevents use of polar NRZ

with direct coupling.

Level transitions in the middle of each bit makes the Manchester-II coding inherently self-clocking, i.e., bit sinc can be extracted directly from the data stream. In addition, the Manchester-II code has no DC component, making this technique suited to transformer coupling. Also, since it is the level transition and not the level itself that is of interest, the signal level is not critical. This simplifies receiver design by eliminating the need for AGC, especially when used with a lossy line configuration, so that all of the described features made this technique the primary modulation method today for aircraft multiplexing (Ref 40:223).

Synchronization. The transfer synchronization method is the fourth issue in the physical/electrical set as defined earlier. The bit and word synchronization must be considered separately in the multiplexed data bus system. A separate clock line can be used for bit synchronization; however, considerations of reliability and clock skew (delay differences between clock and data lines) make synchronization derived from the bit stream particularly attractive. "Self-clocking" modulation methods such as Manchester-II or polar RZ provide clocking information as an integral part of the bit stream, since a transition occurs during each bit time. For this reason, the use of Manchester-II modulation means that no central clocking information need be provided inde-

pendent of the bit stream for data synchronization purposes (Ref 52:24).

The synchronization signal which occurs at the beginning of each word has three essential functions: (1) it signals the beginning of a new word (or message); (2) it identifies the word type; and (3) it resolves the bit synchronization clock phase ambiguity that is inherent in the Manchester data code (Ref 75:24). This word synchronization can be provided by unique bit patterns or unique waveforms in the bit stream. Unique bit patterns are restricted to applications where it can be assured that a particular pattern will never occur in a normal data stream. Unique waveforms may differ from the standard data stream in amplitude or time of transitions. In systems using Manchester coding, the amplitude of the signal is not important; hence, word synchronization can be accomplished by using a nonvalid waveform. This nonvalid waveform is characterized by having only one level transition in three bit times, as opposed to the normal Manchester level transition during every bit. This method is discussed in more detail in Chapter IV (sections describing the MIL-STD-1553 based multiplex buses).

A central clock allows for synchronous operation between all the bus terminal units, but problems of reliability (a failure of the central clock disables the entire system) and clock skew (requiring compensation logic in the terminal

units) make this approach less desirable than the use of local clocks. Although local clocks increase the complexity of the terminal units, problems of clock skew and dependence on a single clock are eliminated. This increases the flexibility of the data bus system, which is particularly important in vehicular systems where growth and modification over the system's life is anticipated (Ref 52:27).

Multiplex System "Analysis Issues"

This set of issues is intended to be a criterion to define the "optimal" architecture and evaluate the attributes of different techniques presented in the next chapter.

The set includes bus allocation scenarios and patterns; bus efficiency and utilization; bus bandwidth (data throughput) considerations; message latency and overhead; impacts of word length and message addressing limitations; capability of a system to transfer periodic, aperiodic, and/or variable length messages, as well as possible block-messages transfer; system integrity and error handling (recovery schemes); functional partitioning and level of isolation; and finally, software interface issues (software and hardware complexity trade-offs and costs). These characteristics will be used -- not necessary in that order -- to evaluate different techniques as described in the next chapter, and to define the characteristics of the "optimal" architecture.

Bus Allocation Scheme. The serial data bus is a resource shared by all subsystems connected to it, and, as already has been said, the method used to prevent simultaneous requests for use of the bus (or to resolve any contention if allowed to occur) is the integral characteristic of the system architecture. This characteristic, namely, will determine to a great extent whether the system belongs to the centralized, federated, or distributed multiplex system architectures, that will be discussed in the next few paragraphs. However, it is necessary to distinguish bus control and bus allocation mechanisms (Ref 20:46) -- bus control provides control mechanism to establish the orderly transfer of data from sources to sinks, while bus allocation associates the "how" (method) and "when" (timing) of message transfers.

Some inconsistency is noticeable throughout the literature in how different authors have defined the conditions characterizing the three types of architectures -- there are several systems that are classified by different authors as belonging to the centralized, federated, or distributed multiplex systems, but the authors were referring to the same architecture! For the purpose of this report the definitions for centralized and distributed multiplexing systems from the reference 49:111-112, and for the federated systems from the reference 20:48 are accepted, as follows:

The conventional, centralized data multiplexing systems are characterized as having:

- Centralized, master/slave control;

- A small number of processors (often more than one only for redundancy);

- A larger number of data sources and sinks (typically sensors and actuators), many being quite simple;

- An interconnection (e.g., bus) structure for connecting the data sources and sinks as I/O devices to the computer(s), sometimes through multiplexers;

- Low complexity communication interface unit hardware;

- A high ratio of application I/O data to multiplex system control information on the interconnection path(s).

Federated control is normally associated with synchronous bus control which is allocated by the master computer. The master computer contains a master executive which is responsible for the intertask synchronization and communication external to an individual processing element (PE). The individual PE need not know the location of any external task since all of the control is through the master and all communication requests must be directed to it. Time sequencing is also controlled from this central controller. In each PE there must be, however, a local executive which is responsible for the coordination of tasks local to a PE; the sophistication of the local executive should reflect the

size of the processing elements and the complexity of task interactions.

The list of important characteristics of a distributed processing architecture is:

- Decentralized, cooperative control;

- More processors (on the order of 10 to 100 in future aircraft);

- An interconnection structure (e.g., bus) for connecting the processors;

- Higher complexity communication interface unit hardware;

- Data sources and sinks that are interfaced to processors (perhaps via data multiplexing subsystems), not directly to the subsystem interconnection paths(s);

- A lower ratio of application I/O data to system control information on the system interconnection path(s).

Distributed systems are coming into increasing use throughout the digital processing field because of the flexibility, performance and reliability advantages which they offer (Ref 46:1) -- examples of benefits in flexibility are (1) the ability to route computing tasks (or bus control, when necessary) to the most suitable processor (as contrasted with local processor that may not be very efficient for a given task), (2) the ability to add processors incrementally as the computing load increases, and (3) the ability to

introduce technical advances gradually, one processor at a time, while retaining existing computers on-line, thus avoiding the major software and systems problems that arise when dedicated computers are replaced, new are added, or some are removed from the system. Performance is enhanced because distributed system bus allocation schemes generally require less overhead than those in the other techniques, thus increasing data throughput, as well as throughput of computing tasks by allowing any temporarily idle computer to be utilized for sharing the load at a busy site. Similarly, reliability is improved because the possibility of single-point failure (in the bus controller) is removed and other processors can be utilized to take up the load of a failed processor until it is repaired. The reliability advantages of distributed systems translate directly into survivability, perhaps the most valuable attribute in any avionics system, so that the price paid for by greater complexity of "smart" bus-interface units is undoubtedly well justified.

In contrast, the primary disadvantage of the command-response multiplex system philosophy is the fact that absolutely nothing happens on the bus without express permission or invitation of the bus controller and then only in compliance with a strict and rigorously enforced set of rules (Ref 90:264) -- any desired flexibility in data rates, data flow, data selections or variations as a function of mode must be

resident in the bus controller. In addition, a significant amount of bus capability is consumed by the bus controller to command the remote terminals. On the other hand, the obvious advantage of this philosophy is due to the fact that the most complex subsystem is often assigned to be the bus controller because it has the most capability as well as the most need to have information from other subsystems, so that the approach ensures that central processing is being done with timely data by interweaving the sequence of bus transaction with the computer processing requirements (Ref 52:34).

In decentralized control schemes (often called "contention" protocol), all participants on the bus are usually given control of a small segment of the traffic, normally that part which is relevant to the controlling terminal. Most contention protocols are source oriented, that is they are based upon transmitting data that is available, not upon receiving data from another participant; however, this is not always true since in some systems a terminal can contend for bus time to request data that is required. There are three basic types of contention systems: Time Slot, Polled and Pure Contention (Ref 90:264-265).

In a Time Slot Contention system each terminal is assigned a fixed time interval during which it is allowed to transfer data or control bus traffic. This protocol is very rigid and it requires absolute synchronization of terminals.

In a Polled Contention system there is a system of "Offer" and "Acceptance" based upon some pre-established criteria. This system is semi-rigid, can be implemented in several ways, and can utilize a variety of criteria to accept or reject the bus when offered. Polling of the bus participants can be accomplished in a pseudo-random fashion by a master polling controller, or can be passed from terminal to terminal in a fixed pattern (round robin).

In a classical Pure Contention protocol, each participant on the multiplex bus is allowed to have free access whenever the bus is not being used by another system (more details are given in Chapter IV, section "Passive Bus System"). This is a very flexible system since it allows traffic to flow based upon need, not upon established rates or times. Several variations are also possible, but all are based on the "if-bus-not-busy" philosophy.

A different concept, which also allows contention to occur but with a very interesting resolving scheme (the "Transparent Contention" technique), has been developed recently by the Flight Dynamics Laboratory, WPAFB, Dayton, Ohio. The technique is discussed in length in the next chapter (pp. 122-135).

An alternative to multiplex bus structured systems is the so-called switched network communication system (also discussed in Chapter IV, pp. 111-117).

Bus Efficiency and Utilization. Bus efficiency is a measure represented by the ratio of data bits transmitted to the total number of bits transmitted (Ref 3:B-1), while bus utilization is defined as the ratio of the total traffic on the bus to the maximum allowable traffic (Ref 3:B-13).

These two characteristics, when combined with message latency considerations, give a good measure of capability of the architecture.

Data Throughput and Bus Bandwidth. Data throughput is a measure of the rate at which a system can exchange data among its terminals (Ref 49:204). As defined, this rate is primarily a function of the transmission medium utilized and its inherent transport delay (the bus bandwidth), the number of simultaneously active data paths, the amount of overhead imposed by the protocol, and, to some extent, the characteristics of the bus interface unit as seen by the avionic subsystem software.

The dedicated, point to point transmission of digital information at rates above 10 MHz is not anticipated in the future aircraft systems. The limitation is based on the constraints of signal skewing due to transmission delays encountered in aircraft installations. The use of an optical transmission medium for the transfer of digital information at or below 1[illegible] MHz is in general not justified since the same function can be performed by a cheaper, lighter weight

electrical interface (Ref 22:47-48), excluding audio, video and radar signals.

On the other hand, the rapid advances in optical technology and LSI circuits make the use of higher bandwidth media an attractive solution. Along the psosibility to include audio, video, and radar signals in standard multiplexing techniques, the use of the higher bandwidth media allows increased intelligence in the multiplex subsystem communication interface units. Namely, this combination of bandwidth and communications interface unit intelligence is highly synergistic (Ref 49:1) -- the bandwidth can be used to achieve significant architectural advantages of distributed processing networks, but the intelligence is required to use that bandwidth; intelligence can be used to achieve significant architectural advantages, but this requires high communication bandwidth.

Another class of advantages which can be expected from higher bus bandwidth is lower avionics subsystem software cost. One example (Ref 49:88) is that the higher bandwidth permits increased communication rates among software modules in different computers. Contemporary software engineering principles recognize that proper partitioning of software can significantly decrease the cost of software checkout, verification and maintenance. Because of bus bandwidth limitations, current practice is to partition software along

lines of minimum intermodule communication. However, the software partitioning techniques which have proven to be most beneficial tend to result in rather high intermodule communication. Thus, they have been largely restricted to multiprocessor systems where the shared main memory provides high interprocessor communication bandwidth. With fiber optics, distributed systems may begin to take advantage of the software cost savings resulting from these partitioning methods.

Anyhow, the bus bandwidth is at a premium in systems with many processors, as stated in Ref 55:21 describing the transparent contention system.

Message Latency and Overhead. The definition for message wait time, or latency, is accepted from Ref 81:21 as the amount of elapsed time from when a message becomes ready to transmit in one device to the completion of receipt by the second device.

Obviously, overhead imposed by the protocol cannot be avoided in any multiplex system. The synchronization bits, interword and intermessage gaps, bus allocation bits, acknowledgments (including status words), message headers, checksums, etc., are the overhead.

Word Length. Digital words that are used to transfer information in avionic systems include a variety of formats: from complex groupings of bits down to a single bit repre-

senting the state of a discrete switch position. For this report, the definitions of types of digital information that need to be considered for avionic subsystems are accepted as follows (Ref 22:56):

Control Word - digital words that require one or more actions to be taken by the recipient;

Data Word - a digital word that contains information relevant to the operation of the recipient;

String Message - a group of digital words which contain information relevant to the operation of the recipient, but for which individual words contain no useful information;

Block Message - a group of digital words where each word contains useful information, but the entire group of words is required in order to be relevant to the operation.

The data word length, defined as the number of bits in the word, determines the accuracy with which the processor can readily perform any calculation. When a word length of "n" bits is used, a number can be represented to an accuracy of one part in 2 . If the processor is not to degrade the accuracy of the system, the word length must be consistent with the accuracy of the transducer data entering the system. In general, avionics data is seldom as accurate as 0.1% (Ref 13:290), and so 10 bits would be adequate. The customary 16-bit information field in the data words is found in most of today's avionics multiplex systems, allowing further accuracy (including for round-off errors in calculation).

The actual data word length is often 20 bits long and includes timing synchronization and word parity, in addition to the useful 16-bit information field. Once determined in such or similar way, the data word length determines the length of the other words in the system (control word, status word, etc.) due to the necessity of unified words in given system.

Message Addressing. A message can be marked by the address of the source and/or destination(s), or may contain the "message name" recognized at the recipient site. The number of available bits for message addressing, as well as the addressing method utilized, impacts the system's flexibility and capability to accept subsequent growth. The source and message addressings are oriented toward broadcast messages, but destination addressing is oriented toward single-destination messages (Ref 49:125-129). These facts affect the system extensibility: to add or remove a message source with destination addressing requires no changes in other subsystems; but with source addressing, a source change means changes in all other subsystems that receive messages from that source.

On the other hand, a system in which the source of a broadcast need not know the number of destinations is more extensible than if it must know, and it is even more extensible if the number may be one (i.e., if broadcast is implic-

it). When explicit broadcast to an unknown number of destinations is employed in conjunction with messages addressed to destinations by "logical names" (i.e., software-defined, versus specific hardware unit "names"), then hierarchical addressing is possible. Hierarchical message-addressing selects a class of destinations based on the structure as well as the content of the address. For example, suppose that destination addresses are four hexadecimal digits each and that a zero digit is interpreted as "don't care". Then the names can be assigned such that all four-digit names without zeros are those of (or within) individual subsystems, while those with one or more zeros are names of groups, as shown in Figure 12.

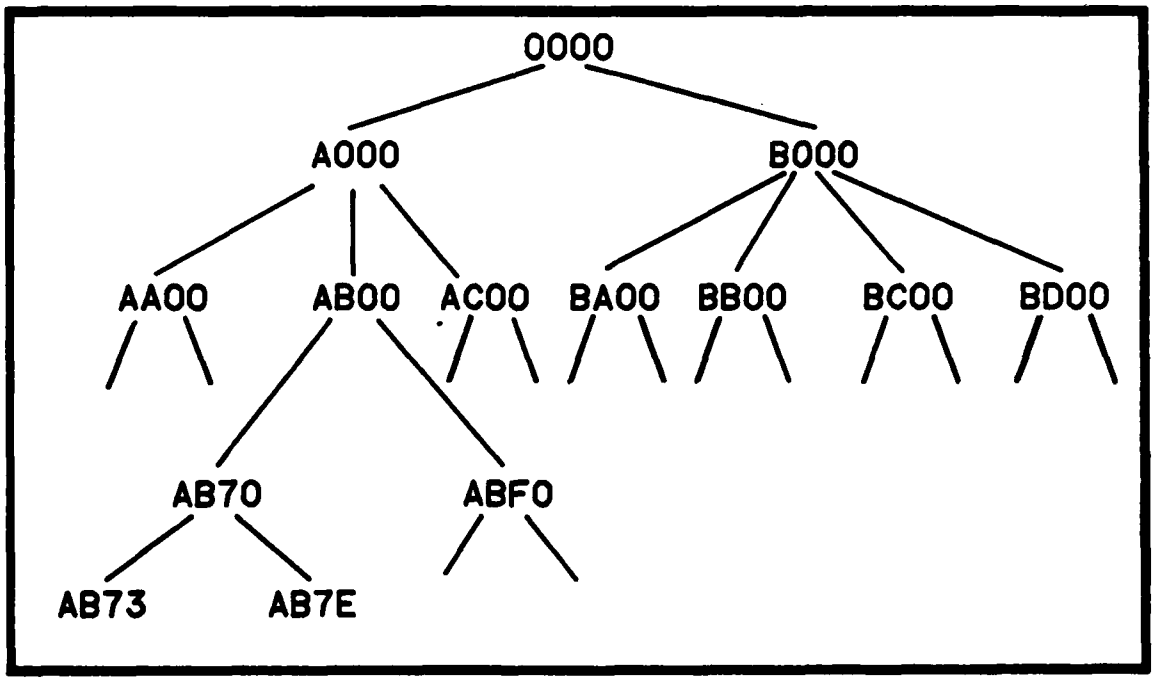


Figure 12. Example of Hierarchical Message Addressing (Ref 49:126)

Thus, messages addressed to 0000 are received by all subsystems, and those addressed to AB7E are received by it alone.

The obvious advantage of the hierarchical message addressing is that it matches the hierarchical structure of the avionic subsystems (Ref 49:125-129).

Periodic versus Aperiodic Messages. In spite of the fact that changes in the avionics data are highly aperiodic in nature, most multiplexed avionics systems today operate on fixed schedules of data transfers. This is, probably, the consequence of the widespread use of command/response protocols that inherently require scheduling of sampled-data messages that occur at regular time intervals. Handling of unscheduled (aperiodic or "trigger") messages in such systems sometimes requires an extensive consumption of the bus bandwidth due to polling of the terminals.

The requirements for the scheduling come from the examination of the largest and smallest minimum iterations and allowable latencies, as described in Ref 3:3/14-3/16. The slowest iteration rate, which is the least common multiple of the faster iteration rates, is normally defined as the major cycle (Figure 13).

Over the course of a major cycle, all periodic transmissions occur at least once and all periodic computations occur at least once. Some exceptions do exist if the itera-

tion frequency is very low (such as Kalman filtering once per 6 seconds, for example).

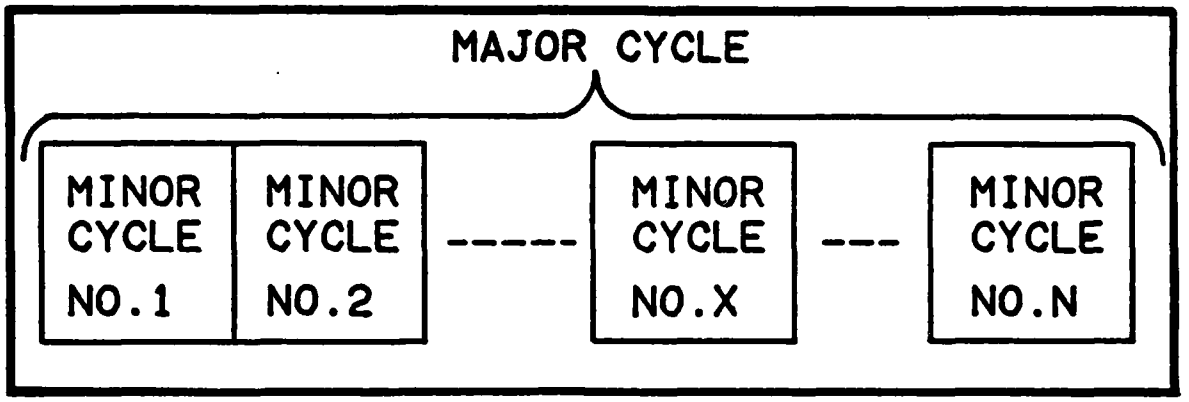


Figure 13. Major and Minor Cycles (Ref 3:3/16)

The minor cycle is normally the frequency of the most rapidly transmitted periodic data. Typical major cycles are about one second in length, while minor cycle lengths can be either binary (2/ sec) or decimal (10/ sec), with common values being 1/128, 1/64, 1/50, etc. (Ref 3:3/14-3/16).

Aperiodic message (time-critical or "trigger" message) is normally assumed to be a very important component of the bus traffic, although such messages are rare and comprise a small percentage of the total message traffic. A weapon delivery message or threat warning message are examples of such messages.

In a command/response system the bus controller polls the terminals at periodic intervals to determine if a terminal has such a message ready to transmit. If there is one, a

communication link will be set up in the manner normally used for periodic message transfers. The procedure is repeated as long as the terminals have these interrupt messages to be processed. Then, the bus controller returns to the regularly scheduled transfers. Obviously, the procedure introduces significant overhead in the message traffic. For example, to achieve the average delays for aperiodic messages of about 1 ms, the bus controller would have to scan all terminals at a rate of 1000 times per second just to look for interrupts alone. Since other periodic message traffic has to be accommodated as well, this rate could be burdensome (Ref 80:2-9).

The attempts made in recent years to design multiplexed avionic systems with transmission of only data that has changed since the last update (the "information-only transfer networks", Ref 61:690), resulted in significant improvements of data latency, providing a meaningful contribution to bus efficiency as well (Ref 90:430). The distributed control systems involving such a transmission scheme explore benefits of aperiodic nature of avionics information. Also, in such systems all messages are basically aperiodic and handling of any trigger message is the normal procedure rather than an exception.

Block Messages Transfer. Multiplexed avionic systems with a block (or string) message transfer capability gener-

ally demonstrate a better bus efficiency measure than systems without this feature. This is the consequence of the lower overhead required to transmit a message block versus word-by-word transmissions. However, the block message transfer capability requires more complex hardware and/or software support, especially when the messages are of variable lengths. In addition, the longer the message permitted, the higher the probability in that a few terminals can tie up all buses available in the system and prevent other messages from being transmitted, unless special restrictions are imposed by the system's hardware or software.

There is no sound evidence that would justify more complex hardware (thus reducing reliability) and/or software (thus increasing cost) just to have the block message transfer feature. A system that requires software reloading as a part of its recovery procedure is an exception.

System Integrity. For real-time command and control systems such as an avionic system, the integrity is considered to be the most important aspect of a system. The system integrity is defined as the degree to which a system tolerates faults, errors, and failures. Faults are referred to as mechanical or algorithmic defects which may generate errors; errors are items of information which when processed by the normal algorithm of the system, will produce failures; and failures are defined as the events at which a

system violates its specifications (Ref 49:203). Integrity involves confinement, detection and diagnosis, recovery (preventing further damage and restoring a stable consistent system state), repair, restart, and reporting. This attribute, therefore, encompasses those referred to by the terms fault tolerance, reliability, availability, and others.

Note, however, that error handling and recovery procedures are dependent to a great extent on the system's application and requirements.

Functional Partitioning. The primary reason for utilization of multiple-level topologies in today's aircraft is to achieve high level of isolation between flight-critical, flight-phase-essential, and nonessential aircraft functions. Such functional partitioning within a data bus system integration separates the aircraft functions, so there are minimum reactions of other functions based on a possible failure of one function. Obviously, both hardware and software functional partitioning techniques are integral and very important elements of the system integrity. However, the techniques of functional partitioning are treated separately in this report to emphasize the importance of the functional partitioning characteristics of a system.

The Federal Aviation Administration (FAA) believes (Ref 85:2) that the possibility of significant design errors in software based digital equipment is much more likely than

for a comparable analog system. The possibility of undetected software errors results in serious problems when attempting to certify installations of redundant digital equipment to be used for critical airplane functions. Namely, redundant installations do not provide protection from design errors. It is believed that functional partitioning, if implemented from the lowest to the global bus level, could help to some extent in solving such testing problems, and at the same time, raise overall system integrity.

Flexibility. The flexibility feature of a multiplexed avionic system does not include only the provisions for anticipated growth during the system's operational life. Additions, deletions, or substitutions of subsystems connected to the bus(es) are to be normal daily procedures to reconfigure an avionic system in response to battlefield changes. The system ability to accommodate widely varying and changing weapons, thus achieving true multi-mission capability, is the unavoidable requirement today. Distributed, multi-level, hierarchical processing networks with a modular design of avionics show a great potential where flexibility, expandibility, and system integrity are required at the same time. An extensive use of modular design permits distributed approaches to the architecture synthesis. This will allow rapid, on-board verification that the reconfigured system will not endanger safety of the next flight. The resulting

flexibility is expected to allow an aircraft to fly more than one kind of mission in a single day (Ref 74:585).

Software Interface. The software interface is the single largest factor in the impact of data multiplexing and distributed processing on avionics subsystem cost, affecting system performance, integrity, and other factors as well (Ref 49:117). At the same time, even conservative projections show that an increase of an order of magnitude in the amount and complexity of advanced avionics software can be anticipated in the years to come.

To illustrate what such a trend really means, an example of today's software complexity and costs will be presented (Ref 74:576):

Assume a 12000-line avionic program was to be acquired with either 0.5, 1.0, or 2.0 percent of the lines of code in error ("instruction error confidence levels").

The acquisition cost was defined as $2,400,000 (25 lines of code per workmonth), $1,200,000 (50 lines of code per workmonth), and $800,000 (75 lines of code per workmonth) for the respective error confidence levels.

The program scenario defined it as having 10 years of useful life and being installed in 180 processors distributed in the same number of aircraft among 10 tactical squadrons. In a squadron, 10 aircraft make two flights of two and one-half hours duration per day.

The total operating time of the fielded system may then be computed by:

$$1.825 \times 10^6 \text{ hours} = 100 \text{ aircraft} \times 5 \text{ hours/aircraft-day} \times 3650 \text{ days}$$

Using the definition for failure rate, and its inverse, mean time between failure (MTBF), the following equation was applicable:

$$\frac{\text{Errors}}{100{,}000 \text{ hours}} = \frac{(\text{error confidence level}) \times (\text{program size})}{(\text{fielded system operating life})}$$

The 0.5, 1.0, and 2.0 percent error confidence levels gave 3.29, 6.58, and 13.20 program errors per 100,000 hours, respectively. Using Monte Carlo techniques with a normal distribution for error difficulty (two, four and six work-months to fix one, two, and three sigma errors), and an exponential distribution for time of error occurrence, the following table was developed:

TABLE I

Typical Average Maintenance Workloading

Error Rate	MTBF	People	Cost
0.5	30395	1.5	$ 900,000
1.0	15923	2.4	$1,140,000
2.0	7575	4.8	$2,880,000

(Ref 74:576)

The part of the life cycle spent on maintenance for the three error rates are seen to be 27, 55, and 78 percent, respectively. A plot of procurement and maintenance curves, as well as a plot of their composite, is included in Figure 14. This composite conforms to the classic life cycle curve with its saddle point of cost effectiveness. Even when the saddle point of the life cycle curve is used, three people are required for the software maintenance.

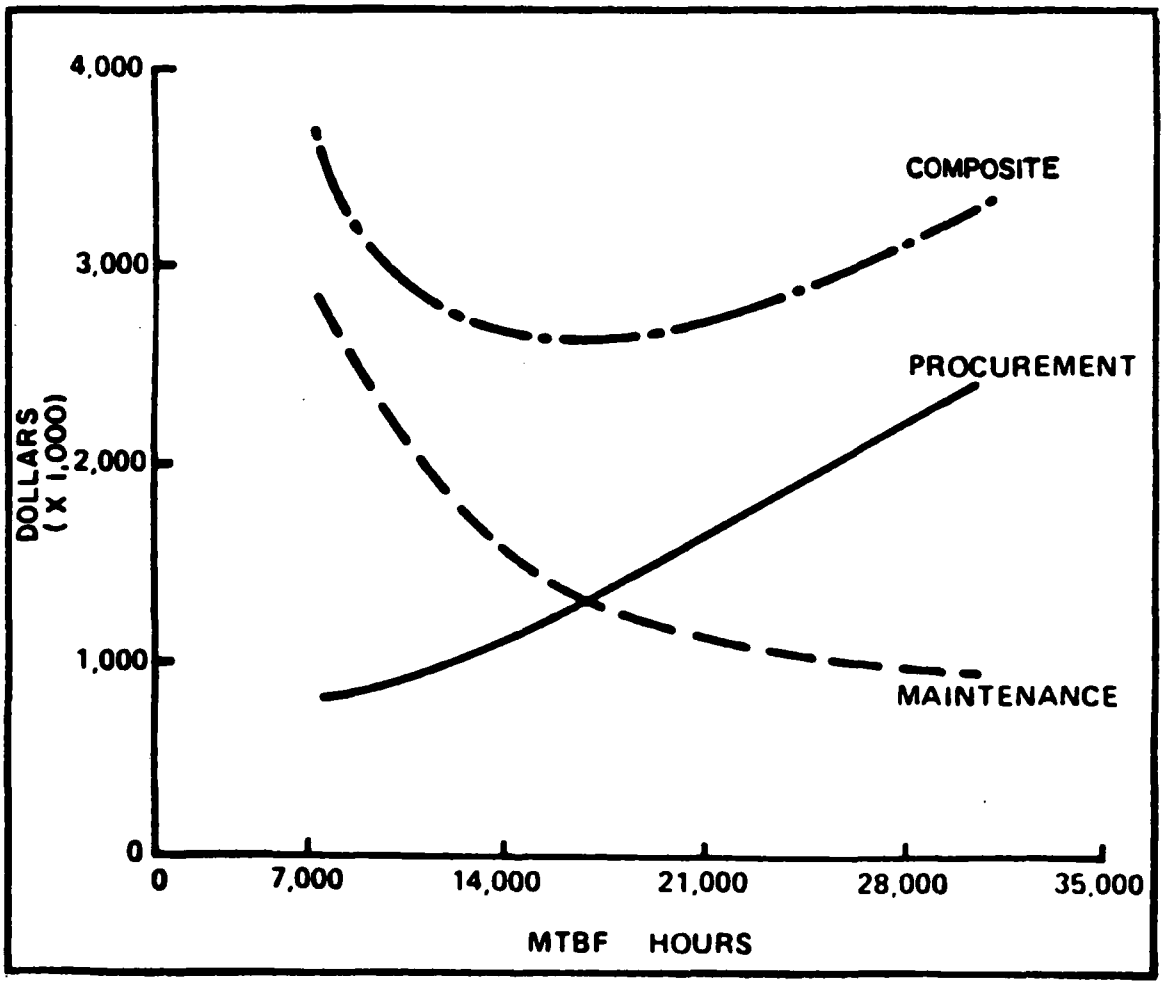


Figure 14. Typical Example of Life Cycle Curves (Ref 74:577)

The operational avionics software problems alone are not within the scope of this study and they are treated only as an integral part of network architectures. In that sense, any software structure is considered in the light of its conformity with the proposed hardware structure. For example (Ref 80:2-5), in the case of a command/response architecture, the design of all aircraft subsystems must be complete before the data bus control program can be completely written and validated. Otherwise, whenever a subsystem change is made during the design phase of the aircraft's avionics, changes must be made in the developing central bus control program, and the entire software program must be validated. Any reconfiguration and subsystem alternations in an operational aircraft will have the same consequence. As shown in Table I (page 54), software maintenance is very costly, and obviously, any system software revalidation must be even more expensive.

In contrast, the software development for subsystems in a contention multiplex system can be done in parallel and the validation can include only testing of the data transaction between a source and a receiver, because there is not exclusive control scheme resident in a single site to be validated. The system reconfiguration has no so significant impact that requires entire system revalidation. This is true especially if data broadcasting is the only means of

communication in the system.

The other side of the coin is the fact that centralized systems, once operative, provide processing with timely data, as already has been discussed (page 39). In flight critical systems which utilize distributed multiplexing techniques, the data transfer synchronization may become the main problem. Software modularization, event-driven schemes, quantized software, data time-tagging, etc., are some of the ways to get around the problem and will be discussed in more detail later.

Hardware/software complexity trade-offs are another important dilemma. A high level of hardware support can decrease software complexity and costs, but introduces hardware reliability problems. A more advanced architecture can offer a highly reliable system made from unreliable pieces (Ref 49:88), but obviously this has a limitation.

Chapter Summary

In this chapter, the current theoretical considerations from the available literature have been used to sort out the essential figures of merit of computer network architectures for digital avionic systems.

Two sets of key issues have been defined: the multiplex system "descriptive issues" and multiplex system "analysis issues". As has been pointed out, the sets are not mutually

independent and the characteristics from each set are discussed with respect to the advantages and disadvantages that they offer.

The characteristics will be used in the next chapter to describe and analyze current network architectures, as well as in Chapter V to define the "optimal" architecture for digital avionic systems.

IV. Avionics Architectures Description and Analysis

In this chapter, descriptions and simultaneous judgements are given on fourteen approaches to the avionics data multiplexing. The choice of a system to be presented is based on either its historical importance or the significance of the ideas included in the approach. However, it is always questionable whether an approach is the "significant" one or not--it depends on the designer's needs and, naturally, each of the technologies does have its own merits for the particular application considered. For complete discussion on some other techniques, the interested reader is referred to the bibliography. (For example, the multiplex systems for F-18 Fighter/Attack Aircraft, YAH-64 Advanced Attack Helicopter, and B-52's Offensive Avionics, are described in Ref 41; the Space Shuttle Orbiter and the Advanced Medium STOL Transport Aircraft multiplex systems are described in Ref 49; the Light Airborne Multipurpose System (LAMPS - a Navy ship and helicopter system) is given in Ref 3; the Advanced Integrated Display System - AIDS can be found in Ref 67, etc.).

The parallel bus systems are not considered as suitable as serial multiplex bus for avionics systems, and will not be discussed in this paper for the same reason, although the parallel bus has the potential for being much faster

than a serial bus. Namely, in a parallel bus system, the number of interconnected wires could be very large and could greatly inhibit the degree to which the subsystems could be physically distributed. Also, a failure in a single wire of the parallel bus would essentially cause the entire bus to fail, while any redundancy requirements greatly increase the hardware and software complexity.

The descriptions and analyses for the chosen systems are kept as short as possible. More data can be found in the reference(s) given at each section title.

The F-15 Multiplex System (Ref 49:13-20 and 44:41-48)

The F-15 is an Air Force air superiority fighter manufactured by the McDonnell Aircraft Company. An avionics data bus system is used to exchange data between various subsystems, most incorporating a digital processing capability.

The multiplex system is unique among the systems considered in that it predates all aircraft multiplex standards. The system architecture is illustrated in Figure 15, which also lists the attached subsystems.

This single-level topology employed in a federated architecture contains a single central mission computer (MC), that controls two sets of dual buses. Backup buses for critical functions are depicted by the dashed lines in Figure 15. Note that flight control is not on the bus, because it

was considered to be too critical to place complete confidence in a multiplex system.

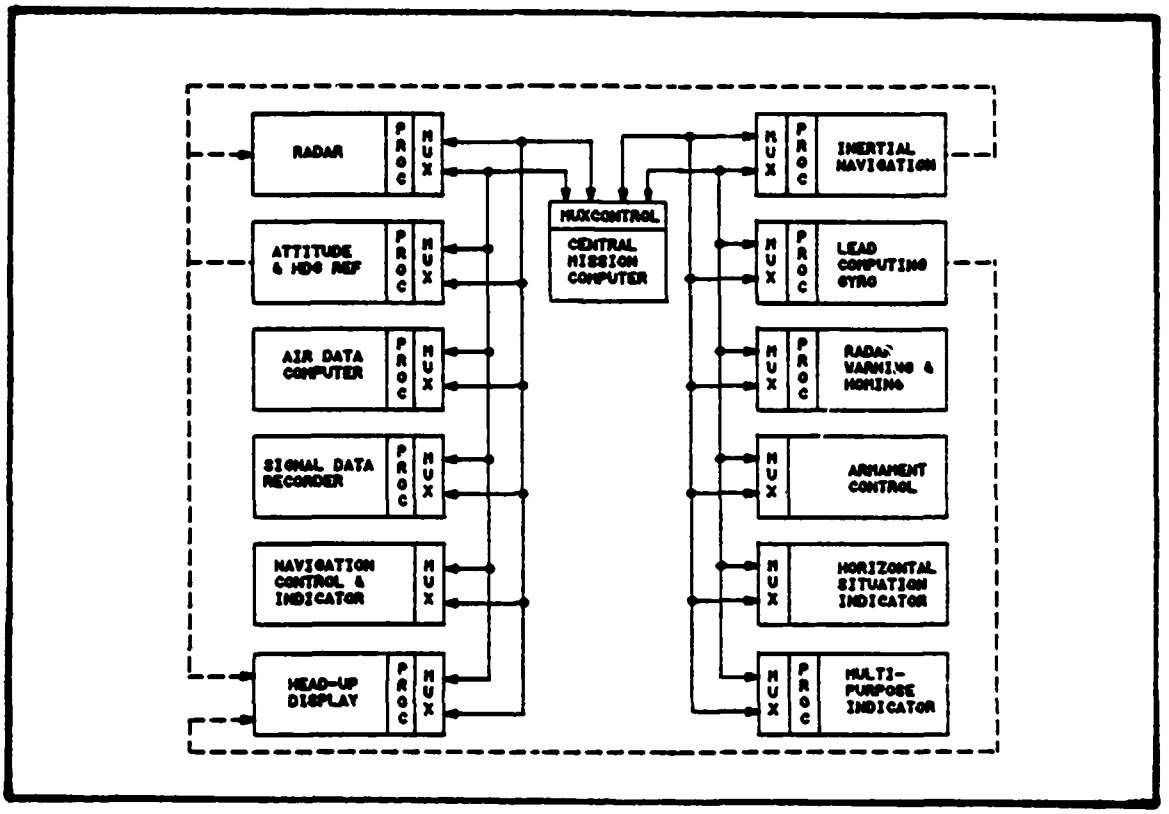


Figure 15. The F-15 Data Bus Architecture (Ref 41:44)

Transmissions are half-duplex (two way but not simultaneous) and are controlled by the MC. A timing clock from the MC is supplied along with the data. The reference clock signal is also used to identify which of two redundant buses is to be used for data transmissions. When the MC wishes to use one of the two redundant buses, the reference clock signal is brought up on the clock line of the selected bus. The two (shielded-twisted) pairs of data and clock lines

constitute a bus, having a characteristic impedance of 68 ohms each. Data is transmitted on the bus at 1 mbs data rate (Manchester-II coding is employed). The signals appearing on the transmission line very closely resemble 1 MHz sine waves since the harmonic content of the data is limited above 1.5 MHz. Data signals, when all either logical one's or zero's look like 1 MHz sine waves. Data signals of altering logical one's and zero's are 500 KHz sine waves mixed with 1.5 MHz 3-rd harmonic components. This produces a wave form with a zero transition which very closely approximates the zero transition of a 1 MHz sine wave. Clock signals look exactly like data signals of all logical one's; i.e., a 1 MHz sine wave.

Maximum line length is considered to be 75 feet as limited by the acceptable phase distortion of the separate clock line in the noise environment of an aircraft installation. The receiving branches of the bus network are terminated by the high input impedance of the terminals (5K ohms resistive component at 1 MHz). No transmission line terminating resistors are used at the ends of the line. Multiplex terminals are transformer coupled to the line in a balanced-to-ground, center-tap grounded configuration. In the transmitting mode, the source impedance of the terminal is 68 ohms. Since the total transmission line length is a small portion of an electrical wave length at the primary frequen-

cy of 1 MHz and frequencies above 1.5 MHz are attenuated prior to transmission, the primary reflections do not produce significant signal waveform distortion.

Message and word synchronization is determined by examining the blank transmission times. More than eight clock periods without data constitutes the end of a message. Words within a message are separated by five clock periods.

Data messages (Figure 16) consist of a select word and 1 to 15 data words, each word containing 16 bits of data and 1 parity bit.

The select word originates only in the MC and performs one of the following functions:

- Request data transmission from specified hardware;
- Identify data to be transmitted from the MC to a specified peripheral;
- Command a peripheral to take some action other than transmit data.

The MC communicates with each subsystem on a one-to-one basis, as specified in the select word address field. Addresses are defined by software contained in the MC. Terminal addresses are implemented in hardware. A control field specifies the message type or command action to be taken. A command indicator bit specifies the select word as a command word, while another bit specifies receive or transmit.

Data transmissions are not acknowledged, but command

messages are. The receiver returns the select word as the data word to the MC when a command is issued.

The word count field in the select word (one for mode commands) indicates the number of words to be transmitted.

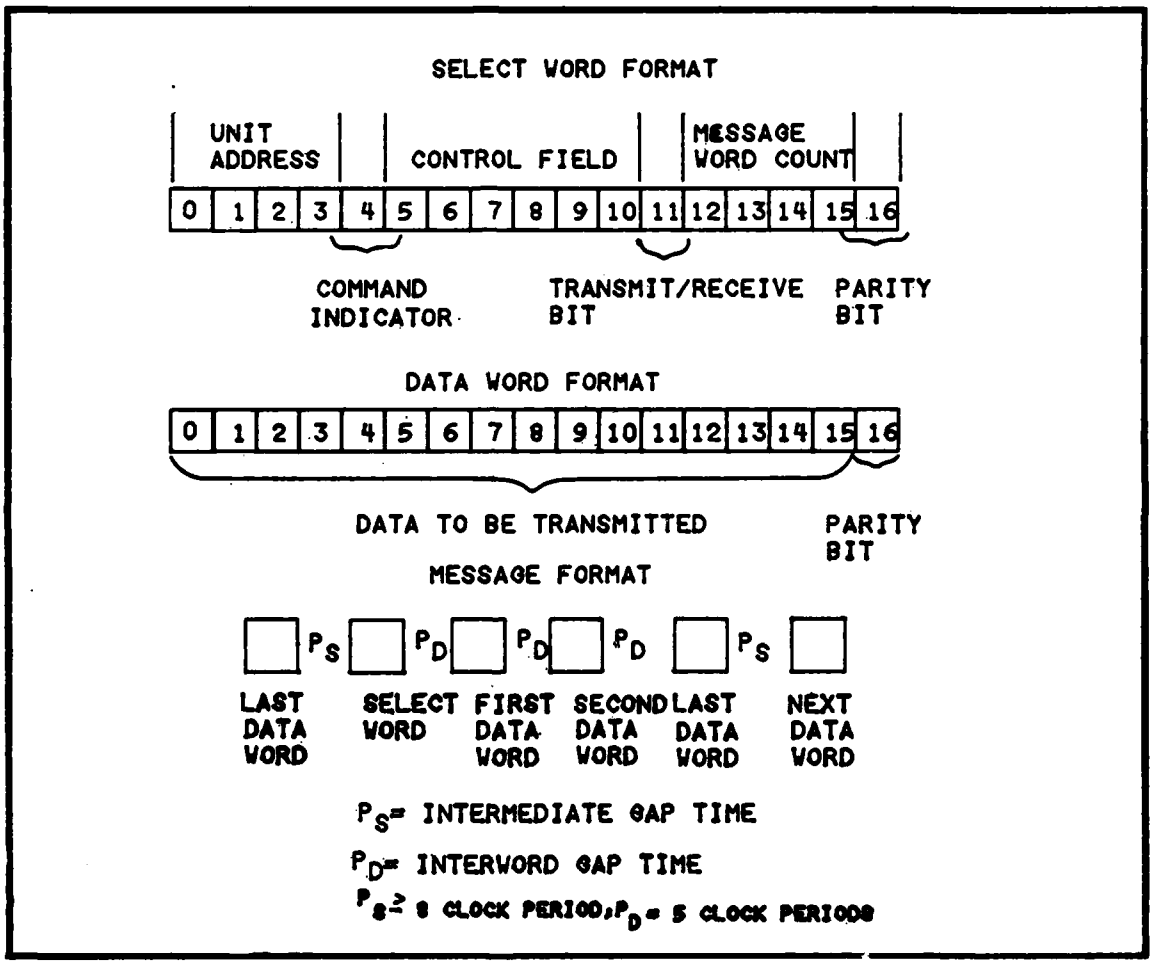


Figure 16. The F-15 Multiplex System Word and Message Formats (Ref 49:17)

Bus traffic is completely controlled by the MC using command/response techniques. The MC performs mission oriented computations and either uses or generates most of the

data on the bus; there is very little requirement for data exchanges directly between remote terminals over the data buses and no transfers between remote terminals are provided at all. If the MC wants to regain control of the bus due to format errors (invalid data bits, parity errors, incorrect word length or spacing) recognized on the bus while a terminal is transmitting, the transmitter will be shot down by dropping the clock signal, which will cause all the RT's to go to the receive mode, resynchronize and prepare to receive a new select word.

A disadvantage of the solution is inherent to the command/response technique utilized: all transmissions on the MC bus are synchronous with allocation via table lookup in the MC software. The consequence is that in order to change the allocation scheme, including the number of terminals, the MC software would have to be modified.

Also, the number of terminals itself is limited by only four bits available for the terminal addresses (to $2^4 = 16$ terminals).

There was no actual data available, but for the purpose of this study the bus efficiency is calculated to be from 13 to 68 percent, depending on the number of data words in a message. It is assumed that the select word, command acknowledgment, parity bits, and intermediate and interword gap times are all the overhead. Bus utilization is around 10

percent (Ref 49:15), while message latency can be up to 2.1 milliseconds, provided that all terminals interfaced to either set of dual buses send full sequences of 15 data words without any command word sent in between (again, the actual data was not available).

The MC has no knowledge of bus usage requests from any terminal.

Synchronization loss, dead terminals, and illegal sequences are all possible bus control errors and will be detected by the receiving terminals. If a receiving terminal detects an error, it ignores the data and resynchronizes to receive a new message. The MC checks word length, word count, interword gap, data dropout (Manchester code violations), and parity. The MC may retry transmission on either the same bus or the standby bus, depending on the past performance. The switch to the standby bus is only upon fault detection on a primary bus and is controlled completely by software.

Subsystems are redundant only in the bus interface circuitry. The interface circuitry is simplified by the added synchronous clock reference at the expense of added aircraft cabling and limited bus line length.

The B-1 Multiplex System (Ref 49:30-35 and 36:83)

Once the data bus concept had been proven on the F-15,

another major USAF system, the B-1 strategic bomber built by the Rockwell International, designed its avionic system around the similar data bus (Figure 17). The B-1's design philosophy was greatly impacted by the MIL-STD-1553 being prepared at the same time.

Three separate bus systems are utilized in the B-1 avionics system, representing a multiple-level topology with equivalent levels of control. Such topology is the primary reason that this system is described here. The first bus, quad redundant, is used for electrical power switching control and management (EMUX); the second is dual redundant for both offensive and defensive mission avionic sensor integration (AMUX); and finally, the third is nonredundant data bus for built-in-test purposes (CITS).

Interconnections between the three bus systems are provided for data exchange purposes only. As with the F-15, flight control remained independent.

The data buses consist of a single twisted shielded pair, with the Manchester-II modulation at 1 mbs rate. Line length is limited to 300 feet. Explicit word synchronization is provided by a three microsecond, 50 percent duty cycle square wave in the same manner required by the MIL-STD-1553 (described in the next section, beginning on page 72).

Data transactions are one-to-one in a command/response protocol similar to the MIL-STD-1553.

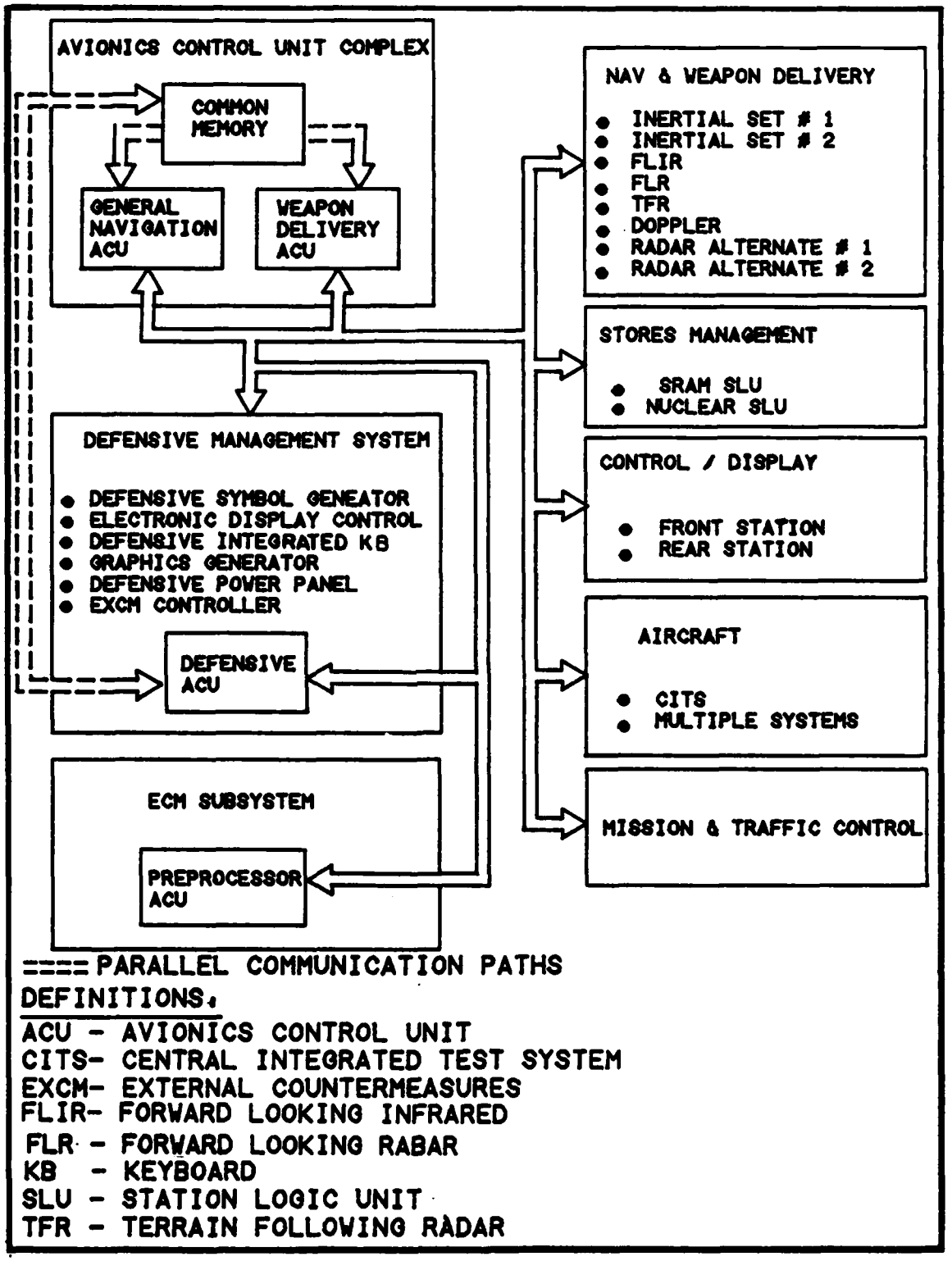


Figure 17. The B-1 AMUX Bus Architecture (Ref 49:31)

In the AMUX (describing here only one out of the three buses -- Figure 17) there are two identical Singer-Kearfott SKC-2070 computers: for the General Navigation Avionics Control Unit (GNACU) and Weapon Delivery Avionic Control Unit (WDACU), connected in such a way that each computer treats the other as a subsystem interfaced to the buses under primary control. Each computer retains a complete executive in memory such that either may assume the full load upon failure by the other. Each ACU has the primary role to control one set of dual buses.

Word length is 24 bits (Figure 18). Data portions are 16 bits plus one odd parity bit. There are three blank bits and one validity bit to indicate the detection of a message error.

The command word contains the subsystem address (five bits) and a 10-bit mode/control and word count field that allows up to 1024-word block transfers to be effected under software control. The subsystem responds with an echo of the command word before transmitting data or after receiving data. Subsystems must use a data word to provide status information to the ACU. Addressing is fixed in subsystem hardware.

Mass memory interfaces to the ACU's through a parallel bus (dashed lines in Figure 17), which requires about a 1.6 megabit transfer rate.

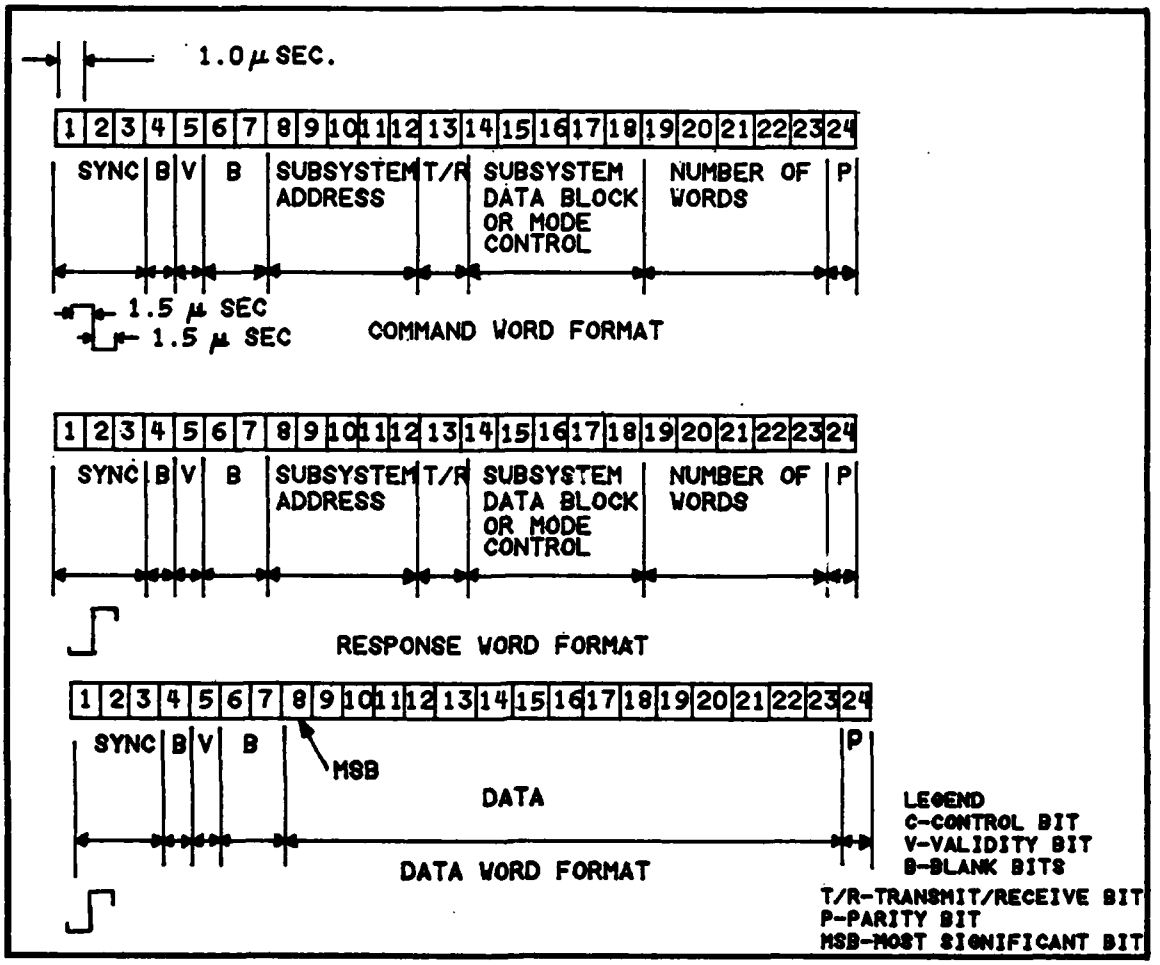


Figure 18. The B-1 Multiplexed Serial Data Word Format (Ref 62:208)

The protocol is rigid as much as in any command/response protocol. Software changes are necessary at the bus control units if new terminals are to be added. A logical limit of up to 31 terminals is the consequence of only 5 address bits available in the command word format.

Bus utilization is up to 45 percent (Ref 49:30) and actual bus loading or response time (message latency) are not

considered critical. Bus efficiency depends on the number of words within a message and is calculated for this study to be from 22 to 66.5 percent.

The advantages of the solution are the possibility for block messages transfer and the fact that in order to accommodate a message with a higher priority, any transmission on a bus can be interrupted by a signal over the alternate bus.

The anticipated errors on the bus (invalid sync code, illegal word or message lengths, parity, invalid terminal address, and data dropout) are all checked by the terminals and by ACU's. Any error that is detected results in no action. When a terminal detects an error and responds with the validity bit set in the response word, or if there is no response by the terminal at all, the ACU will repeat the same message on the alternate bus. If the retry is successful, further transmissions will proceed on this bus until the next failure. Several retries are possible, or the control will be passed to other ACU immediately, depending on the nature of the message. Declaration of the failure is provided, as well.

Failure of an ACU causes the other to assume the entire load. Both ACUs are of equal capability and both have the same programs resident in their memories. Data table exchanges between ACUs are performed each major frame as a normal procedure, thus virtually continuous processing is main-

tained even during the switchover due to the ACU's failure.

The F-16 Multiplex System
(Ref 36:84, 44:15-20, and 49:21-25)

The F-16 is an Air Force combat fighter supplied by the General Dynamics. The F-16 development program coincided closely with the initial publication of MIL-STD-1553 and so became the first vehicle to utilize and flight test a MIL-STD-1553 compatible multiplex data bus system. Due to its extreme simplicity, it is often used in the literature as a descriptive example of the classical implementation of the MIL-STD-1553 standard.

The avionics architecture of the F-16 may be described as a federated multi-computer network (major subsystems have data processing and memory capabilities), although only a Delco M362F computer -- the fire control computer (FCC) is used as a general purpose processor (Figure 19).

The FCC operates as primary bus controller (BC) for the dual buses and sends a separate bus control signal to the inertial navigation unit in case of the FCC failure. The inertial navigation unit is then responsible for bus control, providing a backup system with limited capability.

The flight control function is still absent from the bus. The Westinghouse fire control radar is also separate, using an internal digital multiplex data bus.

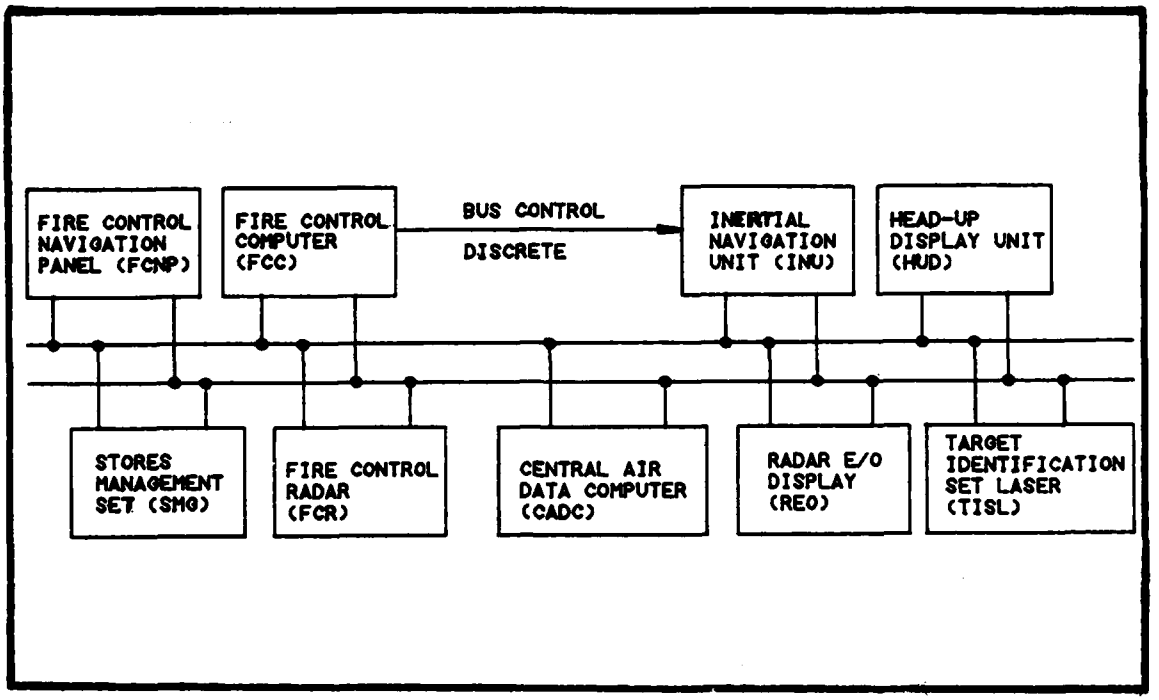


Figure 19. The F-16 Multiplex System Architecture (Ref 41:17)

The data bus itself consists of a dual set of twisted shielded pairs with characteristic impedance of 70 ohms. Although the MIL-STD-1553 allowed up to 300 feet of cable, the F-16 data bus utilizes a short cable assembly (17.5 feet long). All subsystems ("remote terminals" - RT) are attached to the bus by the use of stubs which are connected to the main bus by transformer/resistor coupling networks. Digital transmission is at one megabit/second using Manchester-II code (Figure 11, page 30).

Word synchronization (Figures 20 and 21) corresponds to

MIL-STD-1553. The self-clocking feature of the Manchester-II code waveform is used so that there is no need for a separate clock line (in contrast to the F-15 approach).

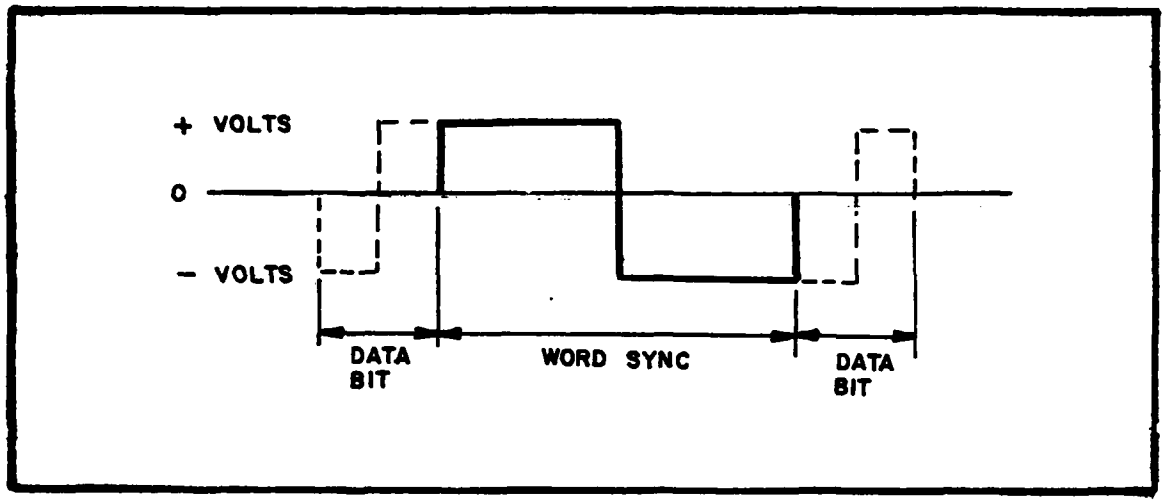


Figure 20. The 3-bit Illegal Manchester-II Sync for Command and Status Words (Ref 65:7)

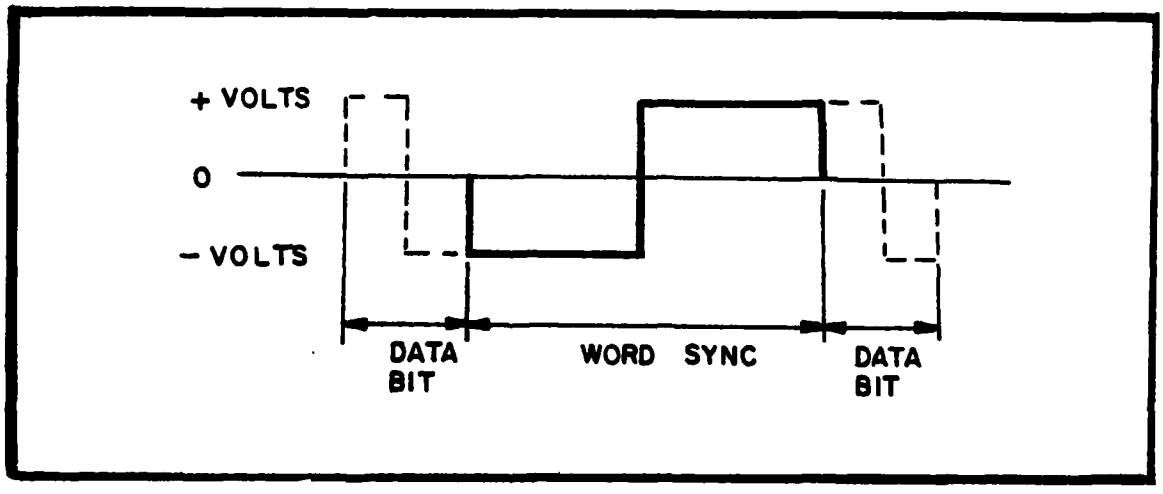


Figure 21. The 3-bit Illegal Manchester-II Sync for Data Words (Ref 65:7)

Three types of words are used to make up the messages which are transmitted between subsystems on the data bus: Command, Data, and Status words, as shown in Figure 22 with the bit assignments depicted, too.

Transactions are command/response with BC-RT, RT-BC, and RT-RT exchanges. A bus controller to remote terminal transfer is accomplished by the bus controller sending a "receive" command word followed contiguously by the data words. After a specified gap time, the addressed terminal responds with its status word, thus indicating proper receipt of the message.

Command words are only transmitted by the bus controller, status words only by remote terminals, and data words by either BC or RT. All words are preceded by a unique synchronization waveform (Figures 20 and 21), which is distinguishable from the remainder of the normally Manchester-II encoded bits.

A terminal to controller transfer is initiated by the controller sending a "transmit" command word to the remote terminal. After the gap time, the addressed terminal responds with a status word contiguously followed by the specified number of data words.

The direct terminal-to-terminal transfer may be affected by a combination of the two previous commands, involving two command words and two status words.

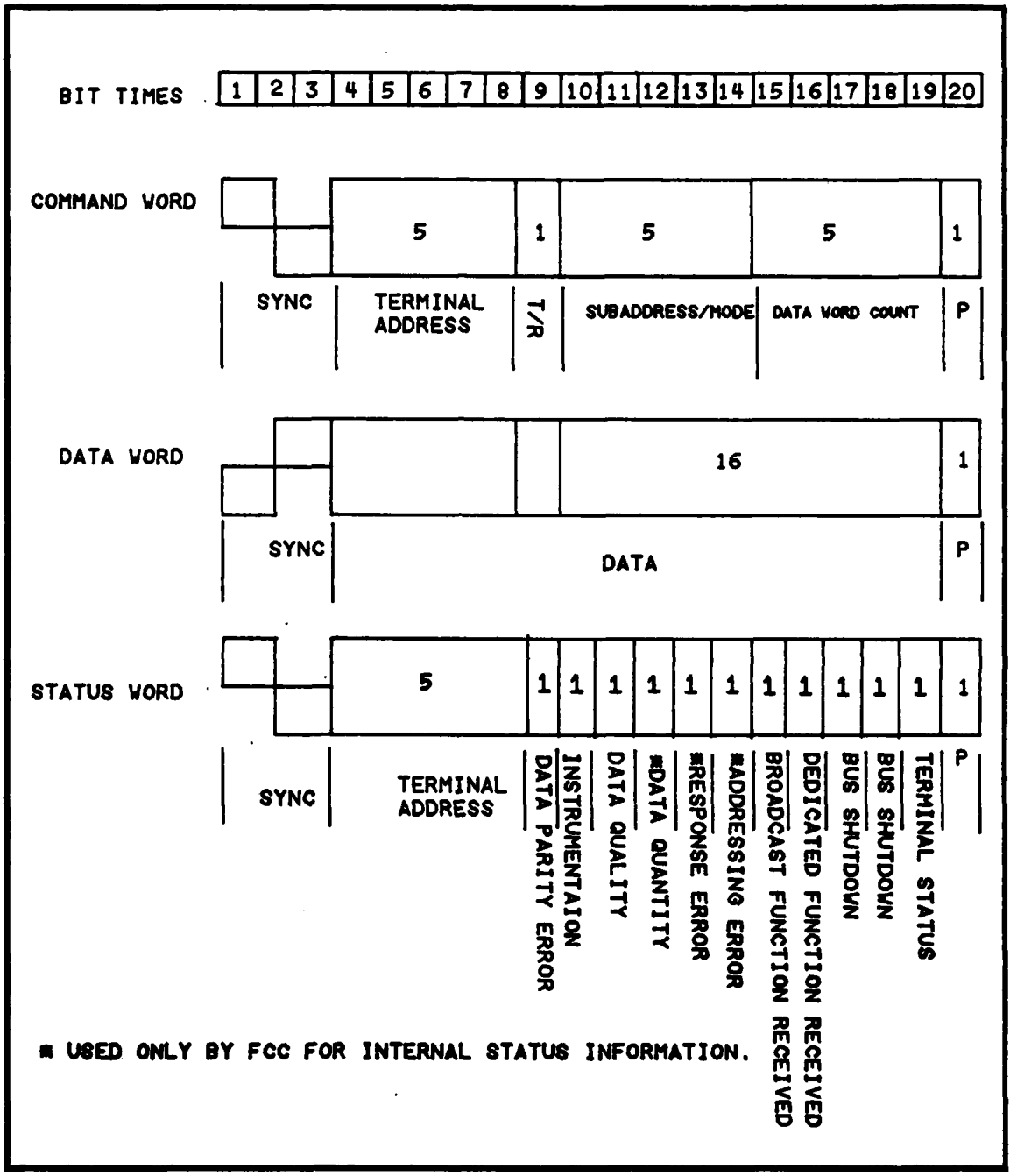


Figure 22. The F-16's Word Formats (Ref 3,Ch.6:6)

Addressing is mechanized in hardware within the remote terminals. The subsystem is capable of receiving a command word on either bus at any time: if a command is received on the second bus while a message is being received on the first bus, the subsystem invalidates the first message and switches over to the second bus. Message length is specified by a word count in the command word. Messages may be from one to 32 words long.

Other subsystems cannot request interrupt servicing except by a status word during scheduled servicing. All transactions are scheduled under software control with variations allowed depending on mode indications.

The bus control algorithm consists of about 1800 words, and the addition of a new remote terminal would require a complex verification procedure to be performed again. As has already been said, such a disadvantage is inherent in any command-response system. Another disadvantage is the 5-bit address field that allows connection of only up to 32 terminals to the multiplex bus.

The 5-bit subaddress field is for the usage of different command modes and message controls as required for given applications.

Bus efficiency varies from 26.6 to 75.3 percent (Ref 3: App B-6, Figure B-1). Bus utilization is about 33 percent in the primary mode and about 10 percent in the backup mode

(Ref 49:23); all response times and data rate requirements are easily satisfied by the one megabit/second bus.

The subsystems' terminals are complicated by the requirement for block message transfer and simultaneous address decoding on both buses. Such provisions, however, allow more suitable scheduling of data transmission and a trigger message priority.

The subsystems check messages for valid sync bits, valid Manchester-II data coding, word length, and odd parity, and report detected errors to the bus controller.

When an error is discovered, the bus controller will take one or more of the following actions based on data criticality (Ref 49:24):

- Take no action;
- Reissue the command over the same bus;
- Reissue the command over the alternate bus;
- Initiate special interface tests for the subsystem.

The broadcast mode, although possible, is not used. The Notice 1 to the MIL-STD-1553B disallows this mode for Air Force use, because the necessary suppressions of status words are not considered reliable enough.

DAIS--Digital Avionics Information System (Ref 3,Ch.6:61-76 and Ref 66:129-136)

Developed as a laboratory program by the Air Force Avionics Laboratory, AFWAL/WPAFB, Dayton, Ohio, to help de-

fine and explore applications of digital avionics and flight control, the DAIS could be used in a wide class of synchronous multiplexed applications. Although this system has not been used outside of the laboratory environment, the concept is interesting as an approach toward viewing the avionics as a whole, not just as a collection of previous defined systems. A partitioning of the avionics system was defined in DAIS such that the mission functions could be partitioned with a goal to achieve maximum utilization of common hardware, maximum sharing of information and resources, and to provide expandable system capability.

DAIS architecture consists of multiple federated processors that provide communications between the processors and the other system elements (sensors, weapons, and controls and displays) through the standardized MIL-STD-1553A multiplex data bus system. The inclusion of the MIL-STD-1553A concept assumes that signals above 400 Hz will be kept off the bus, while the DIAS designers have oriented the system toward the repetition rate of less than 128 times per second.

The flexibility achieved by the concept is primarily due to the use of four classes of functionally standardized "core elements" (processors, multiplex information transfer, controls and displays, and the mission software) that can be integrated in various mixes and configurations to satisfy the specific mission requirements.

DAIS Processors. The number of DAIS processors required is determined by the mission requirements and the mission software to be loaded. The laboratory implementation has only four processors, but this is not the actual limit, as shown in Figure 23.

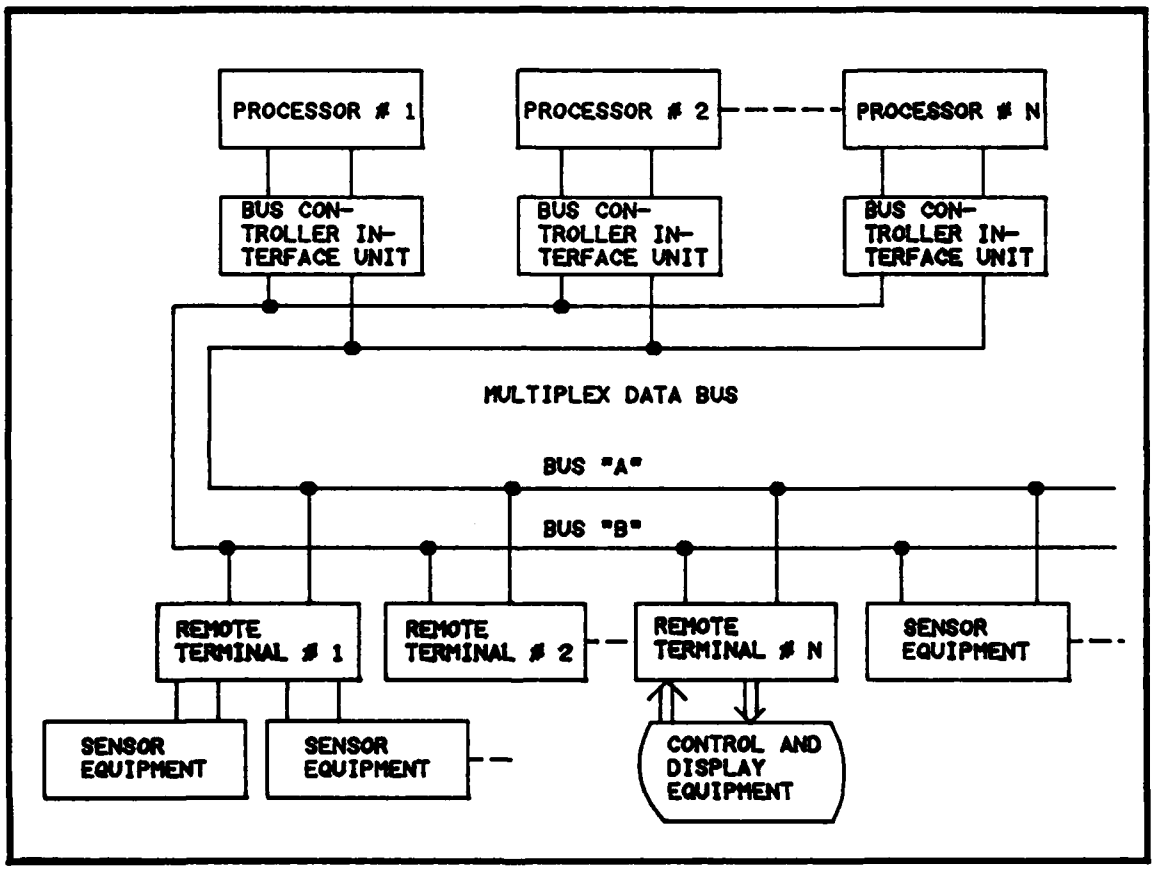


Figure 23. DAIS Architecture - Hardware (Ref 87:131)

DAIS Multiplex Bus. The DAIS multiplex system consists of: bus control interface units (BCIU), one for each DAIS processor; remote terminals which interface the subsystems to the data bus; and the redundant multiplex data buses.

The system control procedures define the bus protocol and error (or failure) management. The BCIU, under control of the master executive, allows bus traffic on only one bus at a time.

DAIS Controls and Displays. This class of the "core elements" is an integrated set that provide the capability for system management by one pilot.

Mission Software. The mission software, as shown in Figure 24 and described in general sense in the references, consists of application software, which performs the processing required for a specific aircraft mission application and subsystems (i.e., navigation, weapon delivery, etc.), and the executive software which performs the real time system control and provides services to the application software. The executive software consists of the master executive, local executive, and backup master executive (if required).

The master executive is responsible for the system bus control and resides in the processor designated to be the master processor. The local executive is located in each processor and provides the real time services, including data read and write, and task control to the application software. The master executive can also reside in a processor designated monitor and provide a backup to the master executive. The executives are table driven to permit flexibility to accommodate various avionic configurations.

The mission software is implemented in the JOVIAL J73/I higher order language utilizing structured programming techniques, standards, and modular software architecture.

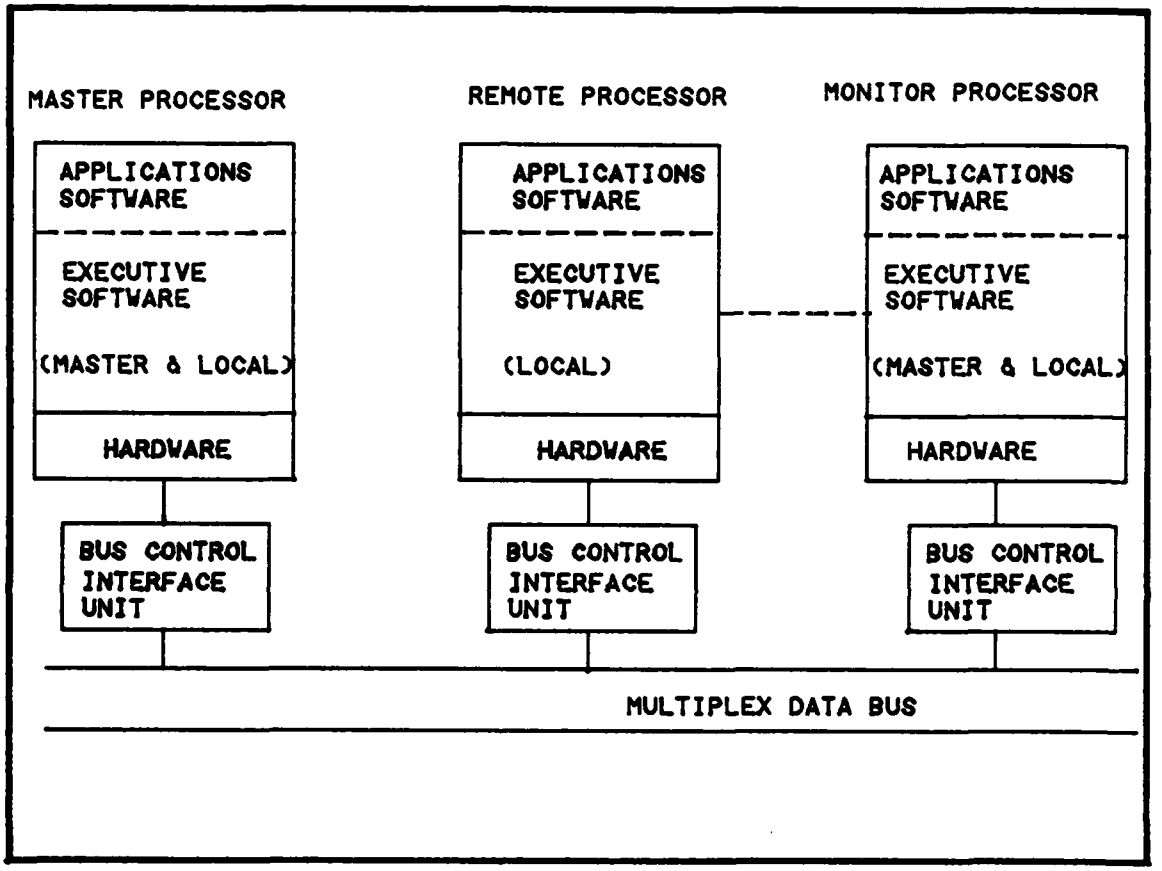


Figure 24. DAIS Architecture - Software (Ref 87:131)

Software is static, but, if necessary, can be reallocated by the reconfiguration procedures provided.

Being a MIL-STD-1553 based system, the DAIS concept retains most of already mentioned disadvantages. However, although under centralized control, this concept provides a hierarchical system architecture with all benefits of such a

configuration. The main feature is the flexibility achieved, fulfilling the easy "mission-to-mission" changes requirement. The architecture is independent of the specific aircraft application and, to some extent, has the technology independence that is desired for any advanced architecture.

DAIS allows both synchronous and asynchronous transactions. Synchronous transactions are according to predefined schedule (a table lookup) algorithm, while the terminals can place requests for asynchronous transactions only through the status words during normal operation.

Mode control and error checking (sync loss, dead terminal and failed bus) are implemented in hardware. The terminals' self-test procedures may be initiated by the master processor, which is also capable to reconfigure the system by switching to the remaining element(s) when redundancy is provided, or by switchover to a degraded capability mode of operation.

According to the Honeywell analysis (Ref 49:216), the DAIS is of moderate integrity, where the centralized, processor dependent allocation scheme is considered as the main disadvantage. In addition, while hardware complexity is not considered high, leaving much of interface responsibility to DAIS's software makes it rather complex and costly.

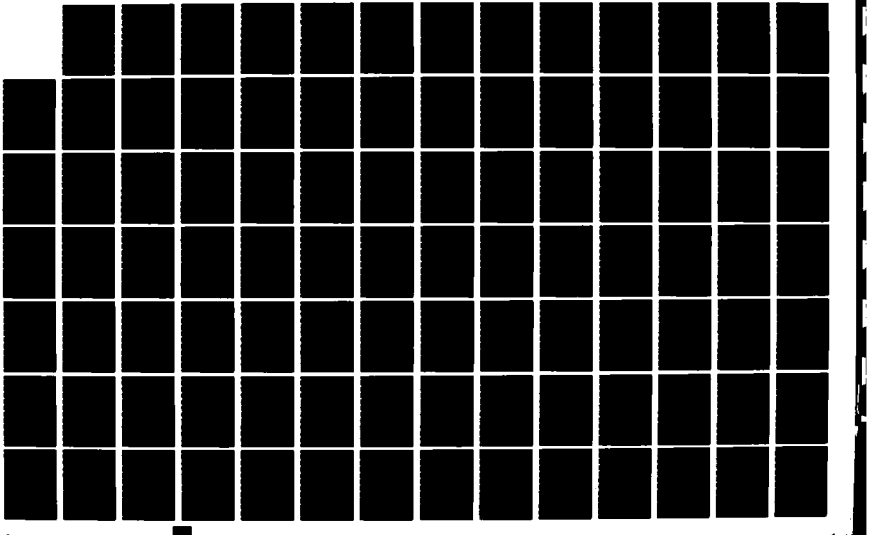
GAMS--General Aircraft Multiplex System (Ref 80)

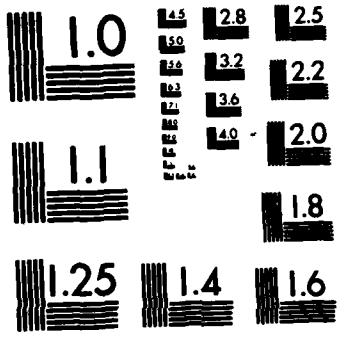
Similar to DAIS, the GAMS is an attempt to define a

AD-A138 151 A STUDY OF OPTIMAL COMPUTER NETWORK ARCHITECTURE FOR DIGITAL AVIONICS SYSTEMS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. M F KRILIC

2/3

UNCLASSIFIED DEC 83 AFIT/GE/EE/83D-36 F/G 17/2 NL





general-purpose, nondedicated multiplex system intended for the asynchronous intra-aircraft distribution of a very wide class of signals, but the concept is based on different architectural and control philosophies than those of DAIS. Proposed as the NADC specification in late 1974, there was no available evidence that the system has been implemented anywhere, except that the A-7E, a carrier based attack aircraft, has been used as a base for some GAMS loading analyses. (The last version of GAMS is known as MIL-E-XXX). However, the concept proposes interesting network structure and this is the reason that it is included in this analysis.

Basically, the GAMS offers a distributed architecture (Figure 25) wherein the transfer of data is independent of any single control source.

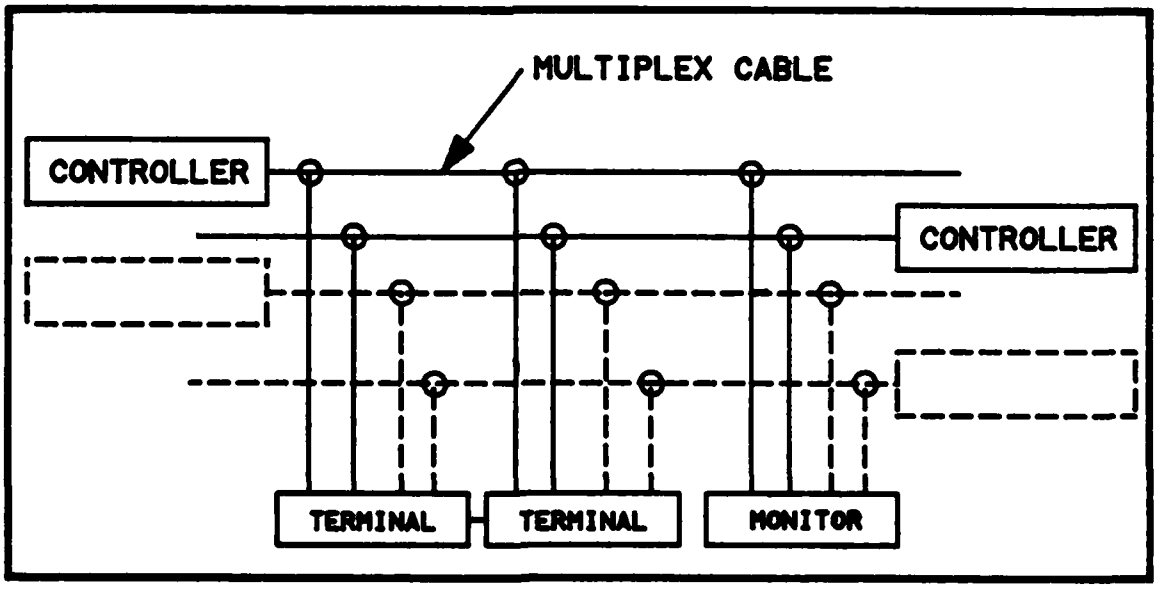


Figure 25. GAMS Architecture (Ref 80:33)

In contrast to a command/response system, GAMS might be called a "request/response" system (Ref 80:2-2), since each user requests bus access when a data transfer is required, rather than being commanded to do so by a central controller.

The major elements of the architecture are:

Terminal. In addition to the functions normally performed by a command/response terminal, the GAMS terminal is also able to participate in a polled system protocol (accept or reject the bus when offered, place a request for message service, etc.). The terminals are modular units with functional partitioning (for example, the "B-function" depicted in Figure 26, page 87, accomplishes all the functions of message initiation, formation, reception, and distribution to user ports). There is a logical limit of up to 32 terminals, imposed by the 5-bit address field in bus allocation words, as seen in Figure 26.

Multiplex Cable. Having up to four cables, each multiplex cable contains two to four bidirectional data buses (the "data channels") and an unidirectional bus (the "control channel") for to the polling function, made by a separate shielded-twisted pairs in a common cable sheath. Each line is terminated by its characteristic impedance (71 ± 7 ohms at 1 MHz) and allowed length is up to 300 feet. The projected data transfer capacity of GAMS is one megabit/second per data channel, so that gross capacity of a full four-times-

four-channel cable installation is 16 Mbps. Minimum recommended is dual-cable installation, but terminals are four-port to support installations with added redundancy or capacity needed. The signal coding is the Manchester-II self-clocking code. Word synchronization is similar to the MIL-STD-1553 specification.

Controller. The data channels within each cable are monitored by separate cable controllers, which offer the channels to the terminals by a polling process as buses become available. No response from a terminal is interpreted as no need for the offered channel. Wait time is 5 data bit intervals plus the maximum one-way cable delay. Each controller is independent of all other controllers in the system.

Monitor. A processor dedicated monitor performs the system health analyses based on BITE data received from the terminals. The monitor is used for in-flight or post-flight fault localization, signalization of the system status and subsequent system reconfiguration (the monitor cannot shut down any terminal or controller, neither can the monitor's failures affect the system operation in any sense).

Bus access is random. A single controller is responsible for its own buses only.

Message formats are depicted in Figure 26 on the next page. Both synchronous (periodic) and asynchronous (trigger) messages transfers are provided.

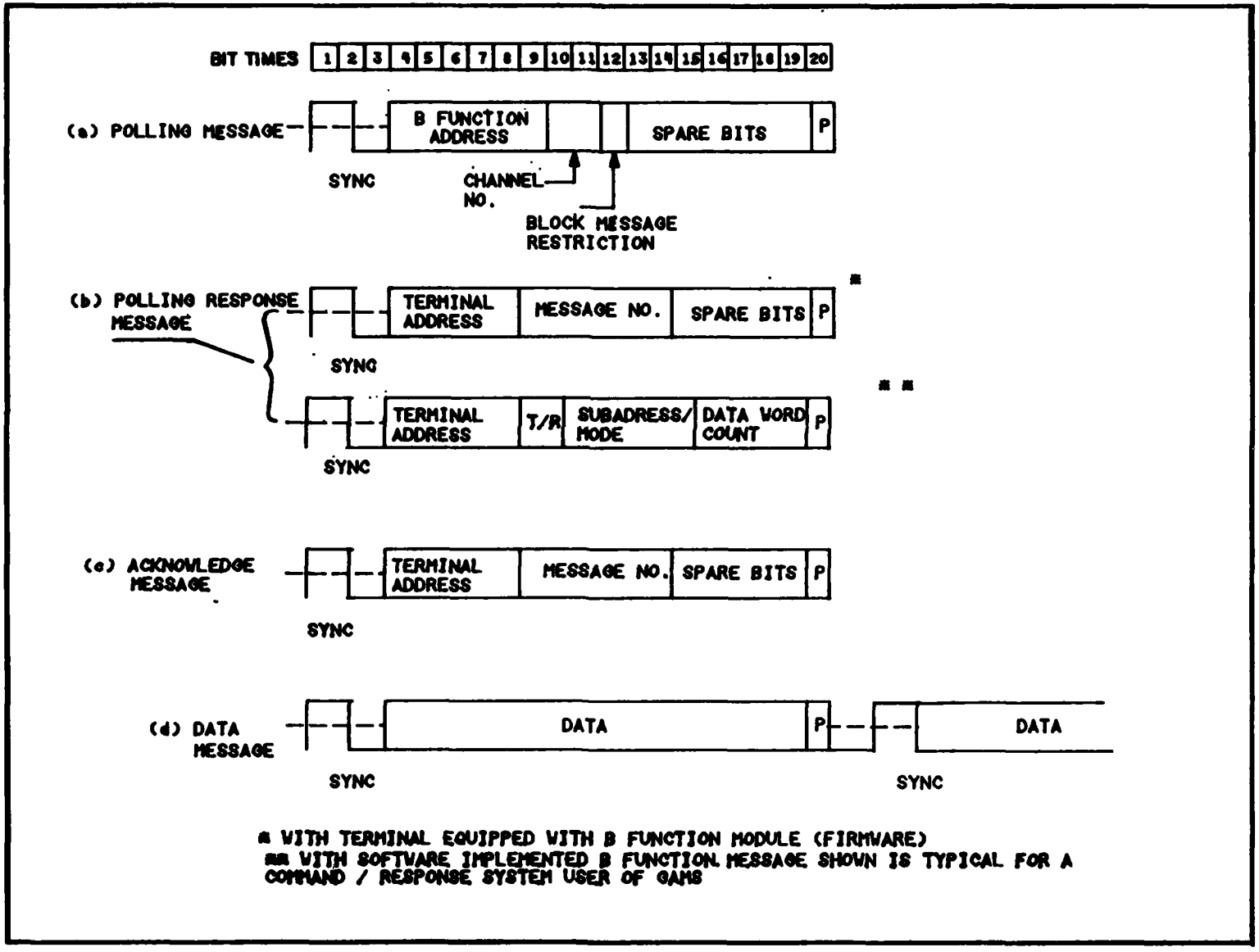


Figure 26. GAMS Message Formats (Ref 80:34)

The immediate offer of non-busy data buses to terminals by a polling process over the control channel is a continuous procedure. Bus access is granted to the polled terminal that has a service request for a message transfer in queue, and the message transfer proceeds.

Contentions between messages are resolved within the terminals, that queue messages service requests on priority and/or first-in-first-out basis.

Messages are transmitted by the source terminal as soon as a link is established. Acknowledgement of message receipt is sent by the sink terminal only if an aperiodic message is sent. Block message transfer is allowed without restriction on the block length, but only if there is at least one additional data bus available in the cable at the beginning of the block transfer. In this way, there is no possibility that few terminals seize all available capacity by long messages transfers.

The distributed architecture, higher data transfer bandwidth, and emphasized integrity are the primary benefits that could be gained from the concept. Data load is shared equally among the system buses that operate independently and simultaneously. Since this is a normal procedure, there is a little reconfiguration needed in case of a bus (or entire cable) failure because the load will be again equally redis-

tributed among remaining channels.

Although the quadruple controller redundancy can be considered as a reliability advantage, it introduces more hardware and weight in the system at the same time. On the other hand, the GAMS controller is a relatively simple device; much simpler than a command/response controller, for example.

Bus utilization and bus efficiency are similar to those of MIL-STD-1553 systems, depending on the particular system requirements (number of data words in a message; number of asynchronous messages that are always acknowledged; command-response, polled or combined mode used, etc.). Note that bus allocation messages and data transfer messages are decoupled. In that sense, the data bus efficiency is 80 percent in most cases, which is the theoretical maximum for a command-response system.

Service requests for periodic messages are originated at the sink terminals. The use of prediction techniques can decrease data latency, inherent to any approach where the user first asks for data and then waits to continue processing after reception of the data. However, when a terminal accepts the offered bus by the polling response message that commands another terminal to transfer a data (or data block), then the effect is the same as in any command/response system, provided that the polling sequence is timed with

the commanding terminal's needs. This timing is assured by the use of clock pulse counters in each terminal and by update rates for information stored in the terminal's firmware.

There is the possibility that a source is busy sending or receiving data over one channel while at the same time it is requested to send or receive data over the other. In such a case, the new request is ignored, causing the demanding terminal to re-request the data. This seems to be the main disadvantage of the approach.

When a faulty message is received, the terminal decides whether to leave its service request flag for the same message, awaiting the next polling offer or to give up the demand for a retransmission of the source data.

Aperiodic (trigger) messages are originated at the source terminals and are transferred at the first polling opportunity. Some delay is introduced by the waiting for the receiver acknowledgment that it is not busy, but the method is still more efficient than asynchronous messages handling in a command/response system (for example, the average delays for aperiodic messages in the GAMS simulation for A-7E aircraft were about one millisecond or less - Ref 80:2-8).

Broadcast mode is not disallowed, but is not supported, as well.

Changing the number or addresses of the terminals will

require changing the polling sequences in each controller and can cause some problems in timing of data transfer. Software interface is of moderate complexity and is functionally partitioned.

HXDP--Honeywell Experimental Distributed Processor Multiplex System (Ref 49:77-85,185-186)

Intended initially for research purposes with regard to applications of distributed processing in real-time command and control systems for small ship guidance, navigation, and display requirements, the HXDP is also being used in a weapon control system (Ref 49:77). The system configuration is shown in Figure 27.

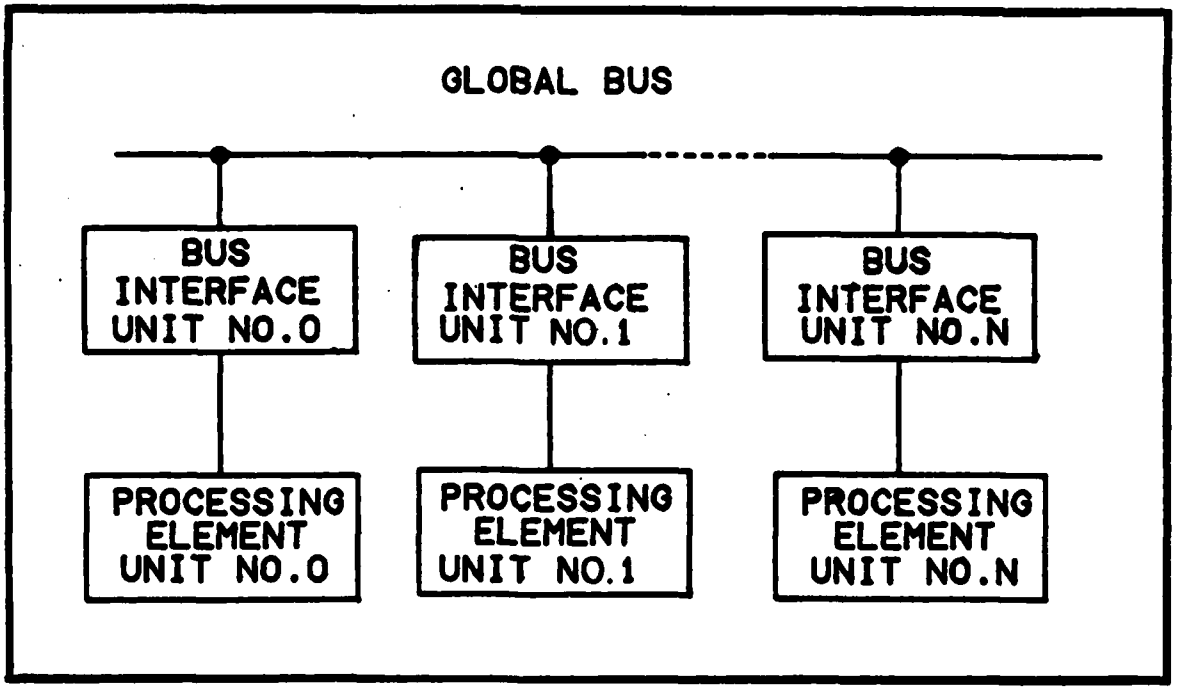


Figure 27. HXDP System Configuration (Ref 49:75)

The HXDP bus is shielded twisted pair. The modulation

scheme is Manchester-II encoding. Synchronization is by message, not by word, using a 5-bit time sync at the start of each message (the SOT in Figure 28).

Messages are addressed to software processes ("destination name" in Figure 28), not to hardware elements. Messages are variable length, specified by an explicit word count field, allowing up to 255 16-bit words plus eight bits of application data in the header (Figure 28). The minimum message length is zero data words (only the header is sent).

All transactions are one-to-one, and are always terminated by an EOT signal. The signal is also used as the bus allocation signal by a specific allocation scheme.

A positive acknowledgment is returned for an accepted message and a negative acknowledgment with a status word describing the problems is returned if the message was refused. No response from the sink terminal is assumed as the negative acknowledgment and retry of the message will be attempted up to three times, when the EOT will reallocate the bus.

A minimum length transaction requires about 100 μsec ("zero" data word, properly received), giving bus efficiency of about 10 percent; a maximum length normal transaction of 255 data words takes about 3.3 msec, which translates in a bus efficiency ratio of more than 98 percent. The longest possible transfer (255 data words with no proper reception

until the third attempt) requires about 13.4 msec, bringing the bus efficiency down to about 33 percent.

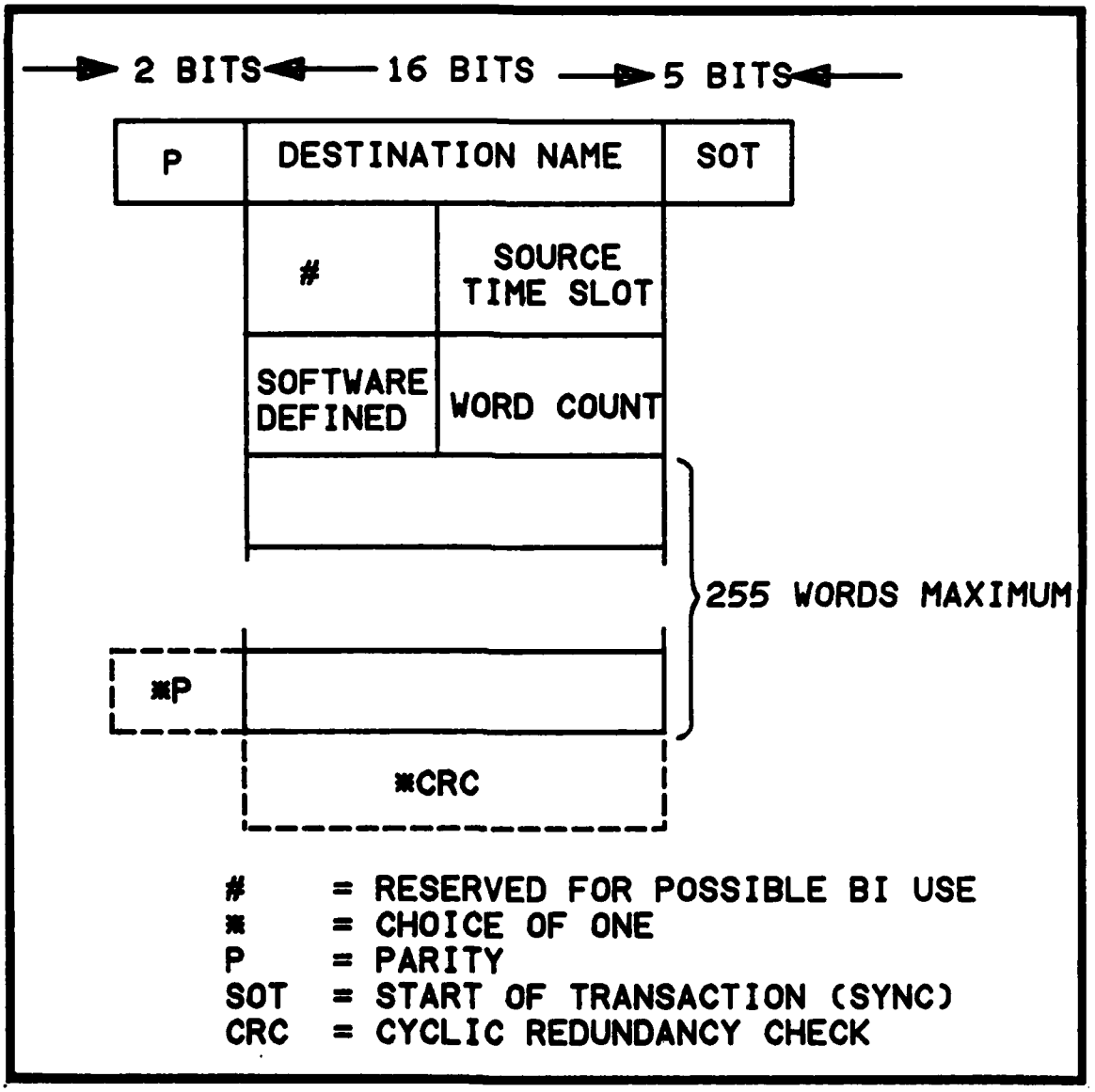


Figure 28. HXDP Message Format (Ref 49:80)

Bus allocation is decentralized and asynchronous (i.e. variable length messages independent), according to the "Vector-

Driven Proportional Access Mechanism", shown in Figure 29.

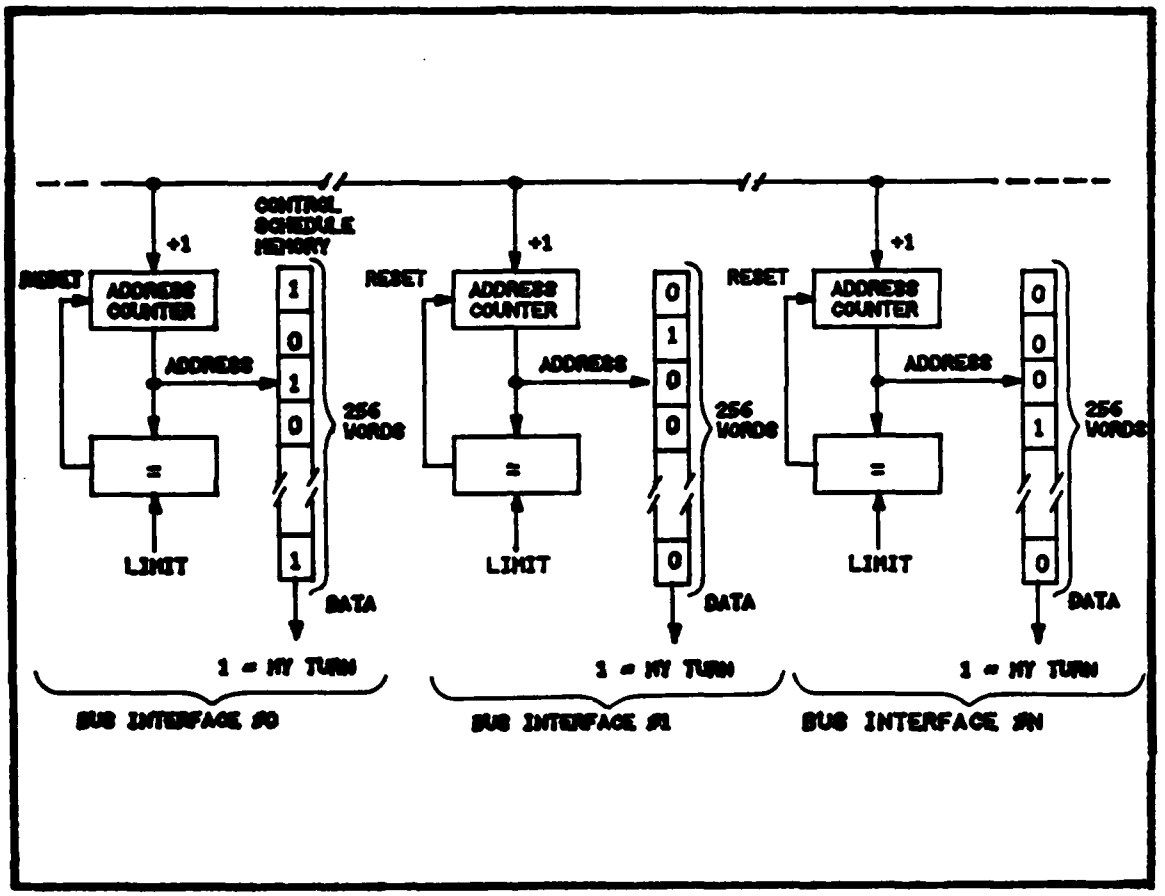


Figure 29. Vector-Driven Proportional Access Mechanism (Ref 49:186)

The mechanism allocates the bus to only one BIU at a time. Each BIU is assigned an explicit circular binary valued allocation vector and an allocation vector index. All BIU vector indices are incremented at the same time, starting from the same point in the vectors as initialized at power-up. The indices increments occur each time at an

allocation signal (the EOT) on the bus. The BIU having a "one" in the position currently pointed to by the index is allowed to transmit. (No two or more BIUs have a "one" in the same bit position in their allocation vector tables). A BIU that has nothing to send at its time, reallocates the bus by sending the EOT only.

In fact, the technique uses a polling-based scheme, where the BIUs are implicitly polled in a predefined order by the table lookup algorithm resident in each BIU's software. In such a way, the explicit offer through separate control signals is avoided, thus significant savings in the message overhead, system weight, and system size are obtained. Also, the system integrity is enhanced in that bus allocation control signals are kept off the transmission bus, thus not being overlapped with data transmissions through the same bus.

These advantages hide a couple of disadvantages:

(1) Any change in number of BIUs connected to the bus would require corresponding changes in the vector tables in all BIUs. This is absolutely necessary if a BIU is added and may require costly changes in the system's software due to changes of the polling sequences for some critically timed messages. On the other hand, if a BIU is deleted, the tables need not be changed -- the only consequence would be wasting some time when the deleted BIU's turn comes. Namely, the

missing BIU will be handled by the procedure provided for the case when a BIU fails to send the EOT to reallocate the bus: the time-counter with the smallest time constant (each BIU has such counter with unique time constant) will time out and generate an EOT reallocating the silent bus.

(2) More serious problems would cause a healthy BIU which, for any reason, did not "see" the reallocation signal orderly sent by another BIU. In such a case, the BIU will not increment its vector index when the others do, and sooner or later, will try to transmit at the time when some other BIU transmits having the right to do so. The problem should be solved by the potentially colliding BIUs: each BIU, during every message on the bus, normally compares the allocation information (contained in each EOT) to its internal allocation data. In such a way, the comparison, plus history data should help the BIU to determine its own correctness and eventually inhibit its own transmitting. There was no exact data available about the efficiency and reliability of such solution tested in practice, but the software is clearly complicated due to the complex collision resolving scheme. In addition, even if the BIU discovers the collision, a message may be lost (or unacceptably postponed) and, furthermore, this can happen several times while the BIU is trying to resolve the problem. In that way, a healthy BIU with potentially important messages in its queue could be

out of the system for a while, just because of a transient error that caused one EOT to be overlooked. It seems that such possibility could cancel potential integrity advantages of the approach.

Loop Multiplex System (Ref 49:146-161)

A loop architecture was designed by Honeywell to emphasize the use of mature or relatively well-understood technology and architectural features, while maintaining some reasonable level of integrity and performance.

The system has up to four loops, all simultaneously active. Each loop interface unit (LIU) can be using only one loop at a time, with the exception that an LIU currently receiving a message is also capable of sending a busy-refusal message in reply to an incoming message on any other loop. The system topology is shown in Figure 30.

All loops are unidirectional, allowing use of point-to-point fiber optics links. All LIUs participate equally in a Pierce loop control scheme (Ref 49:151). The Pierce loop is organized as a collection of fixed-length "packets", or message blocks, which continuously circulate around the loop through two-packet shift registers buffers in each LIU.

The number of packets in the system is defined by the product of the number of LIUs and the number of packet buffers (two) per LIU.

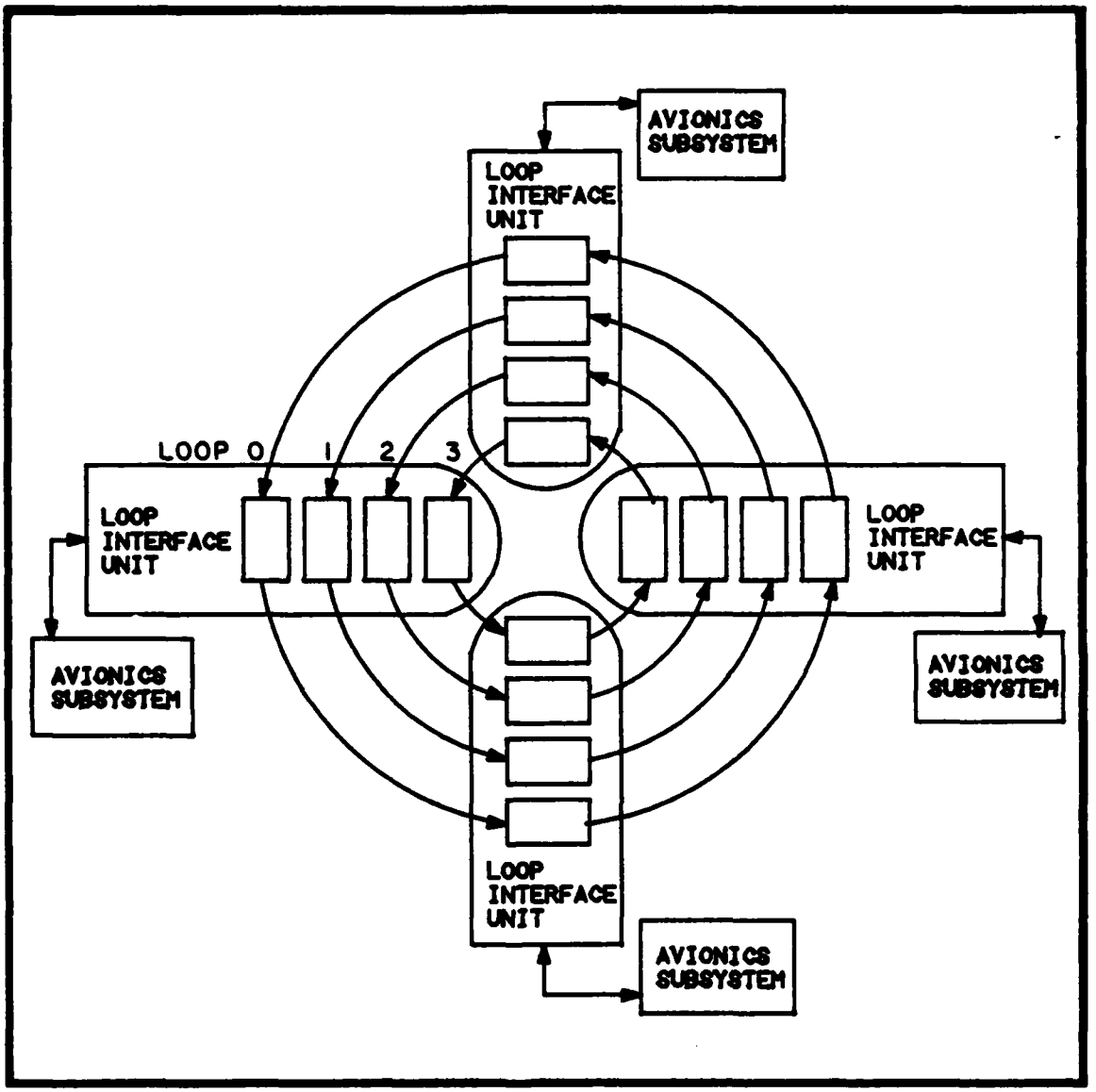


Figure 30. Loop System Topology (Ref 49:147)

The circulation of "packets" around the loop is illustrated in Figure 31 on the next page. Each packet contains a status field which indicates various things about the packet, including whether it is "empty" or "occupied".

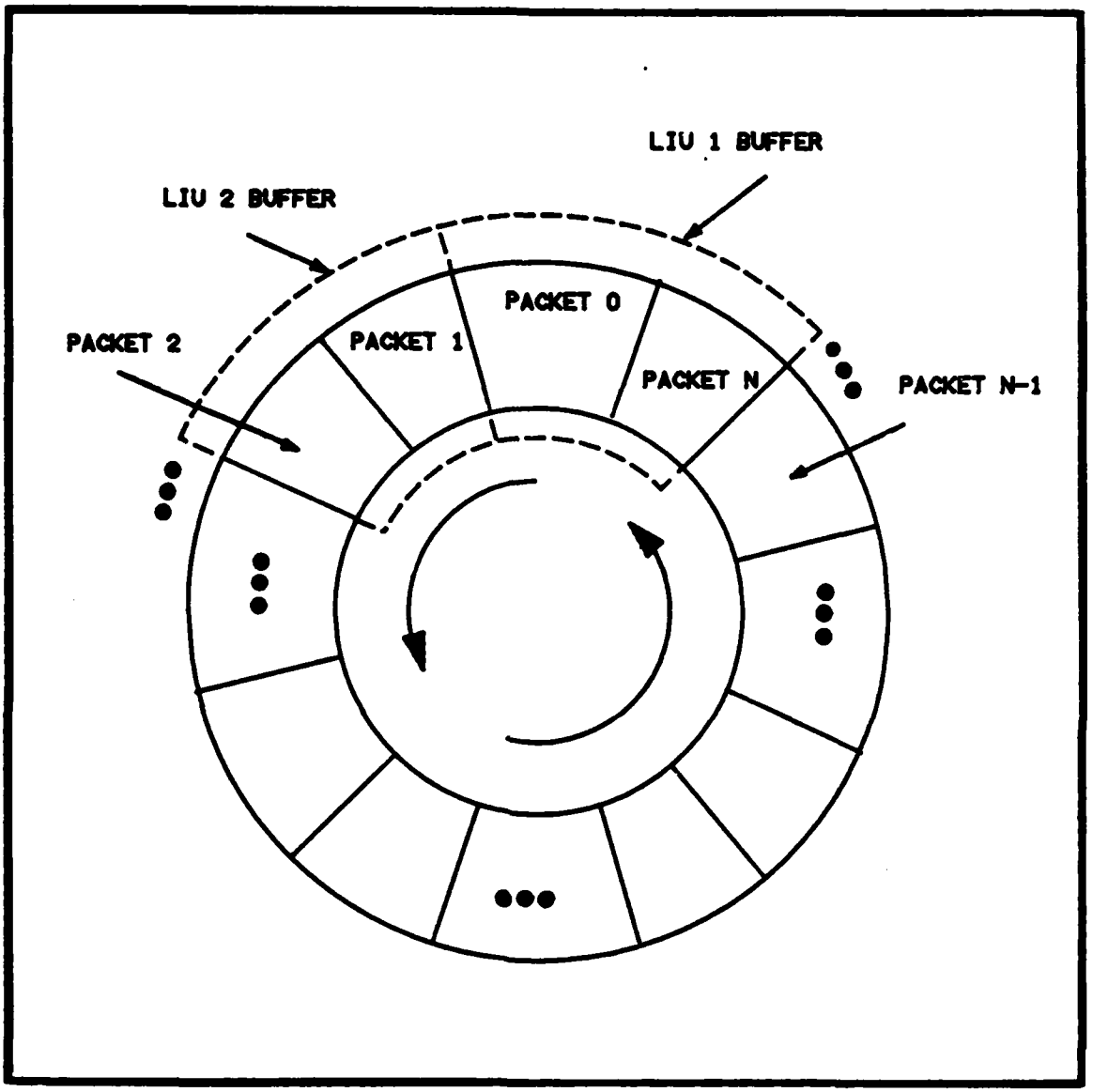


Figure 31. Pierce Loop Organization (Ref 49:152)

An LIU sends a message by first waiting for an empty packet to arrive in its buffer and then writing the message into the packet as it is shifted out of the buffer. The packet is shifted around the loop until it gets to the

destination LIU, which then copies the message contents, sets a flag in the status field acknowledging receipt of the message, and allows the packet to be shifted out onto the loop again. The packet returns around the loop to the source LIU, which then marks the packet "empty", making it available again.

Each word in the packet begins with a synchronization waveform, followed by three bits which specify one of five sync types, followed by a 16-bit word (Figure 32).

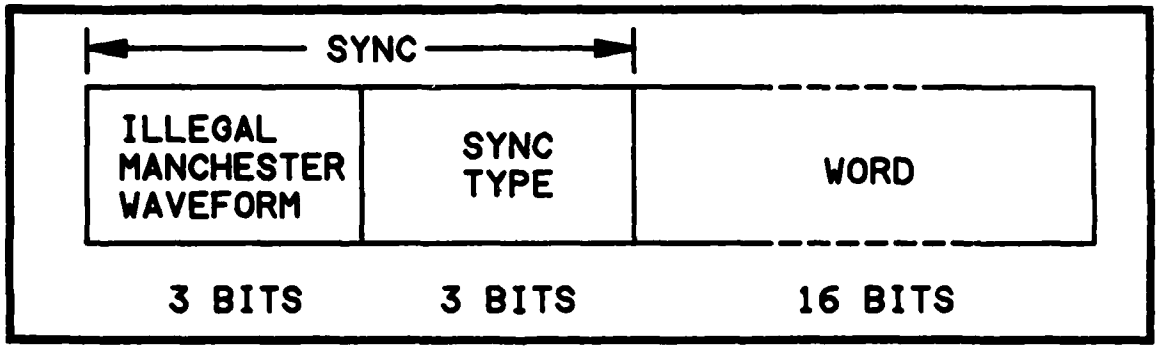


Figure 32. Loop Word Format (Ref 49:156)

Packet format for the loop system is shown in Figure 33 on the next page. As seen from the figure, destination name is the process name to which this packet is intended. The word contains 11 bits of name and 5 bits of single-error-correcting code. The source name is similar. Status/Control bits contains the various flag bits and fields used in the control and management of the packet: one bit is for Empty-Occupied, one for Going to Receiver - Coming from Receiver,

one for Acknowledge/Negative Acknowledge, five bits for Requested Message ID, and last five bits for Single-Error-Correcting Code.

In the fourth word, eight bits are reserved for the word count field or the reason for refusal.

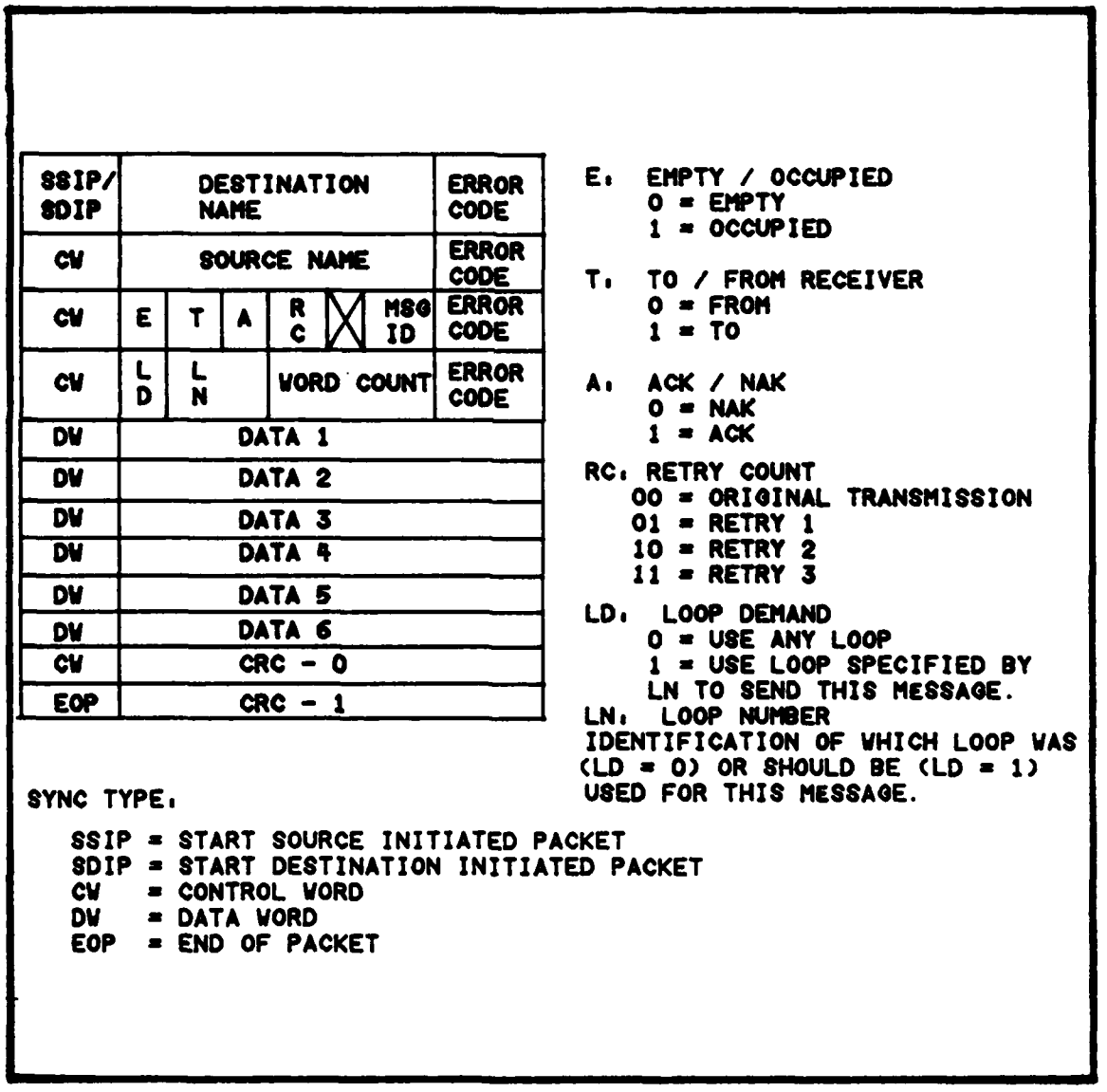


Figure 33. Packet Format for Loop System (Ref 49:158)

There can be up to six data words in the message. Space for all six is always reserved in each packet, but the word count specifies how many of the words are valid.

All packets include two CRC words, providing the necessary low-level probability of undetected errors.

The system architecture is designed, as is said, with the primary goal to exploit "low-risk", well-known technologies and this is the main advantage of the approach. At the same time, much of weaknesses and disadvantages of the technique result from the very same reason.

Bus efficiency is very low - from only 6 percent when one data word is transferred to the maximum of 36.3 percent when all six data words are transmitted within a message. These figures are even less when message retries occur (there is up to three possible retries after the original transmission) or if the fact of messages circulation is encountered. Namely, when a receiver copies the message content into its buffer, the "T" and "A" bits are set to "0" and "1", respectively (Figure 33), but the same message continues travelling toward the originator which is responsible to "empty" the packet. Such method enhances the system reliability, but the data throughput is lowered by the fact that packets are occupied for some extended time after the message has already been received properly. For the same reason, message latency could be higher than acceptable for

avionic systems, while already used packets are passing by and cannot be used until made "empty".

There is no way provided to give any priority to trigger messages. Block messages transfer is not provided. The broadcast mode, although possible, is not supported.

A loop can be used only if all of LIUs connected to this loop are healthy, thus jeopardizing the system integrity by only one faulty LIU. Garbled destination name in a message will not be recognized by the receivers, and the packet may circulate through the loop forever, disallowing the use of the packet for any other message. Garbled source name has similar consequence, because the packet will never be made "empty".

Passive Bus System (Ref 49:161-178)

The system is also designed by Honeywell in a laboratory environment, with integrity (i.e., fault tolerance) as the primary goal. Electrically passive radial arm couplers are used to configure the buses, which are redundant and used simultaneously by BIUs.

The system contains four passive buses in a redundant configuration, as shown in Figure 34.

The buses operate concurrently and independently. Each message carries the identification of the bus selected by a BIU to transmit the message. This helps the BIUs and subsys-

tems test for and identify bus faults. Bus failure causes little reconfiguration: the BIUs simply stop sending and/or receiving on the faulty bus.

Each BIU is able to transmit on only one bus at a time. It can receive on all buses at once for the purposes of detecting a "start of message" sync, looking up the name of subsequent message, and responding with a busy-refusal, but it can receive only one message at a time.

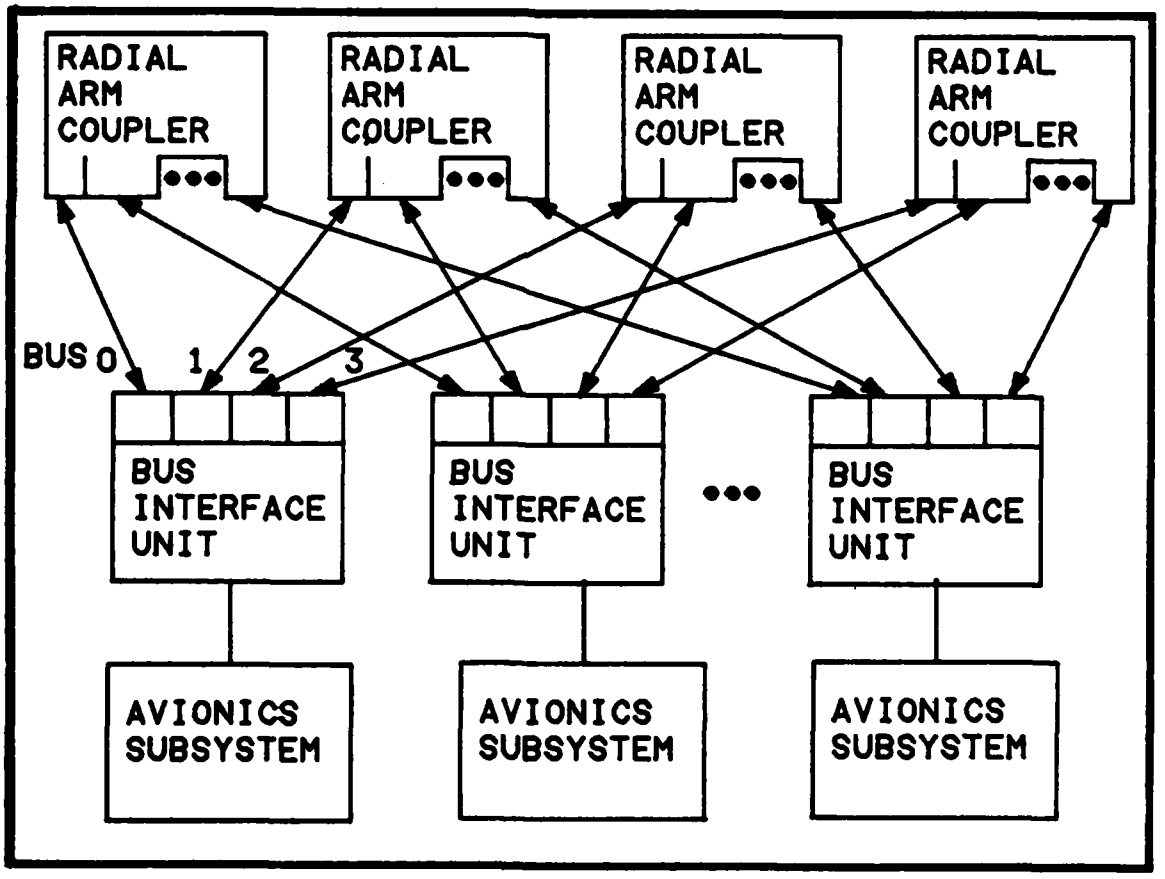


Figure 34. Passive Bus System Topology (Ref 49:163)

Global buses are implemented using fiber optic cables as the transfer medium. Each bus contains a radial arm coupler (RAC) and several arms, with a bifurcated fiber bundle connected from each terminal to the RAC.

There is a sync on every word of a message, rather than just one at the beginning of the message. This improves the accuracy of the demodulation. The Manchester-II encoding is used again.

Bus control is decentralized and distributed across all BIUs. Bus allocation is based on an "Ethernet"-like scheme, which allows BIUs to transmit on the bus any time it is available. Namely, in the event that more than one BIU starts at the same time, a "collision" is said to have occurred and they all stop sending immediately. Each then waits a slightly different interval before attempting to retransmit. The one with highest preassigned priority "times out" first and gains access to the bus, completing its message, while all other BIUs wait until the bus is once again available. Some time is, therefore, wasted while colliding BIUs "time out" and it can be serious when demand for the bus is high and collisions are frequent (Ref 53:310).

Bus allocation is synchronized by the "bus available" (End of Transmission) signal which is transmitted on the bus by a BIU as it vacates the bus. Failure to terminate a transaction properly is detected by a minimum period of bus

inactivity. When this occurs, another BIU will uniquely time out (different time-out constants in each BIU) and send the EOT to synchronize bus allocation.

All messages are transmitted on a one-to-one basis with no third party involved. All messages are addressed to software destinations but, because of integrity goals, both source and destination names are also included. Each subsystem is allowed up to 31 names which will be recognized by that subsystem's BIU.

Only one message is transmitted per transaction. Word format is the same as for the Loop System (Figure 32, p.97), and organization of message formats is also similar enough to be avoided for separate presentation in this subsection.

The positive acknowledgment message format consists of two words containing both the destination and source names. The negative acknowledgment will have, in addition, a BIU status word detailing the reason(s) for rejecting the message. The EOT signal, also, consists of three words: destination name, source name, and the bus allocation number.

The 11-bit destination and source name fields permit up to 2048 global names in the system. Since each BIU can recognize up to 31 names (one is reserved for the BIU itself), there is enough name space for 64 subsystems. Five bits of error detection code on each name will detect up to two single-bit errors. Following the data is a 16-bit CRC

code which is required to provide adequate fault detection over 258 words (there are up to 255 data words in a message).

Although the organization of message and word formats are very similar to those for the loop system, the bus efficiency of the passive bus system is higher in spite of its greater overhead (explicit acknowledgment words and three words within each EOT signal). The bus efficiency maximum value is around 70 percent (a message with 255 data words, one positive acknowledgment and one EOT). The minimum value for bus efficiency is about 7 percent for only one data word transferred, and is close to the minimum value for the bus efficiency of the loop system. The doubled value for the maximal bus efficiency of the passive bus system is the consequence of longer messages permitted.

Block message transfers and/or broadcast mode are not provided because they were considered potentially more complicated from the BIU hardware complexity viewpoint. Even such slight increase in hardware complexity was rejected by the system designers who wanted a system with maximal integrity.

The most valuable conclusion that can be drawn from the approach (Ref 49:216, Table 2.4) is the fact of a very reliable collision bus allocation scheme, which allows decentralized arbitration about the bus usage. This is made by the colliding BIUs, without a third part involved. Although

high risk is unavoidable when such mechanisms are to be developed, the potentially high reliability that can be achieved encourages further research in the field to investigate other possible collision schemes.

Active Bus System (Ref 49:179-196)

The system is another (Honeywell) research program, this time oriented toward achieving high performance (fast response time and high throughput), but without raising subsystem software costs.

The "active bus" system is so called because the interconnection bus is simulated by point-to-point links between adjacent BIUs and electrically active (i.e., powered) repeater logic in each BIU. However, the links do not form a loop. Topology of the system is illustrated in Figure 35.

The avionics subsystems are all attached to a set of bit-serial global buses via special purpose processors (BIUs). The global buses operate concurrently and independently, and a bus failure causes little reconfiguration: the BIUs simply stop sending and/or receiving on the faulty bus.

In a fact, each global bus is simulated by point-to-point links between pairs of BIUs. This avoids the risk, attenuation and limited number of arms which are presented in radial arm coupler buses. To create the effect of a bus, two unidirectional fiber optic links are used between the

pairs of BIUs to avoid the power loss from bifurcation of a single link (it is assumed that fiber links which can transmit in both directions at once do not exist). The modulation scheme is again Manchester-II (BiPhase-Level) encoding.

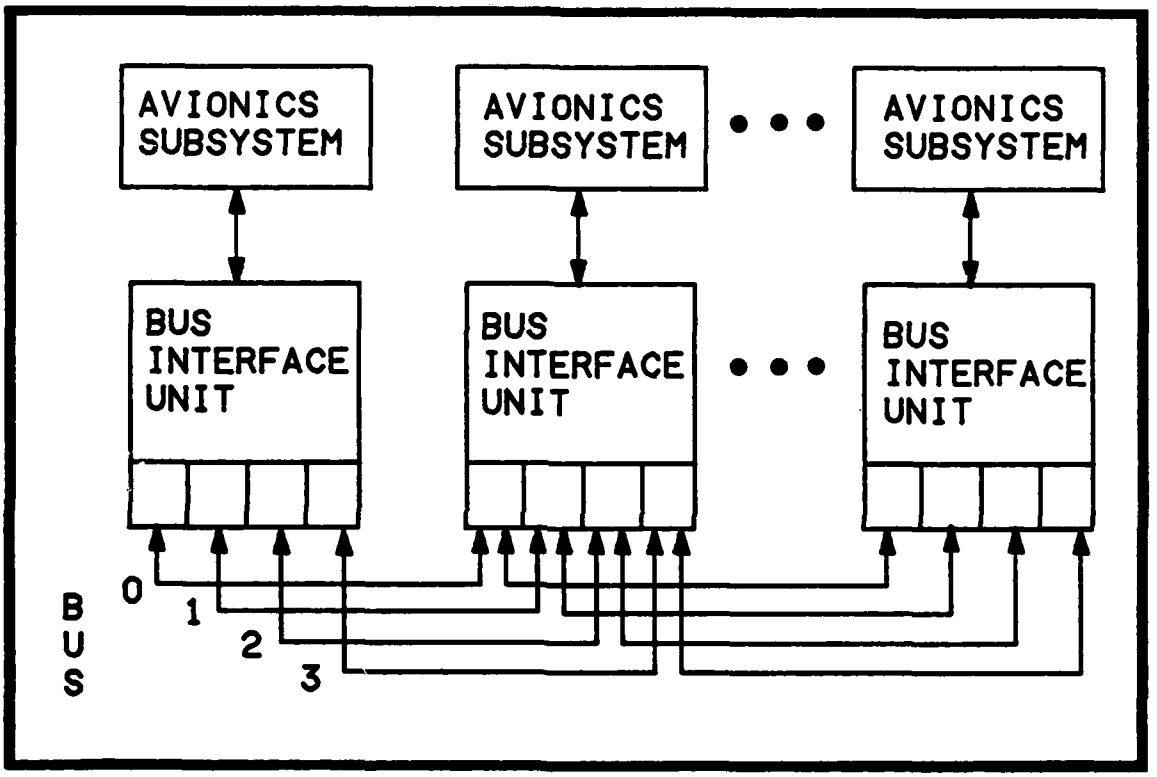


Figure 35. Active Bus Topology (Ref 49:180)

Bus allocation is done in a decentralized manner by all BIUs. Each bus is independently allocated. The bus allocation scheme is the "Vector-Driven Proportional Access Mechanism", already explained by Figure 29, page 94.

Both one-to-one and broadcast messages exist in the protocol. All messages are addressed to software destina-

tions. Broadcast messages are explicit (signaled by a bit in the message).

Only one message is permitted in a transaction, with positive or negative acknowledgments (the later case includes a status word detailing the reasons for rejection). The word and message formats are similar to those already explained (Figures 32 and 33, p. 100 and 101, respectively) and the illustrations will not be repeated herein. Up to 31 different messages can be defined for each source name in a BIU.

Messages are transmitted from FIFO output queues in each BIU. The output queue contains pointers to messages, not the actual messages themselves, which allows rapid insertion, deletion and resequencing of messages on a priority basis. Receiving is done by subsystem memory buffers. There are up to 32 input buffers per subsystem, one for each destination name.

The goal of high performance (i.e., bus bandwidth and efficiency) dictated that the designers wanted to avoid rejecting messages, thus as much decreasing message transfer retries as possible. It led to an interesting solution for the BIU's repeater logic. Namely, if a bit in a message is received as an illegal Manchester waveform, the message will not be automatically rejected. A special circuit, called a modulation correction unit, will arbitrarily convert detected illegal bit to a legal bit waveform. Obviously, with the

50 percent chance the bit will be correctly restored. If not, there is high probability that the error will be detected and corrected by the use of the message CRC word. The net result is that the message has not been repeated and bus efficiency as well as data throughput is enhanced.

The implicit polling by the use of the Vector-Driven Proportional Access Mechanism (Figure 29, p.94) introduces possible weaknesses that have already been discussed. In addition, the mechanism has the disadvantage in that it depends on the subsystem transmission needs being accurately estimated a priori (at system configuration time). This has also been discussed earlier (page 56).

In normal operation, bus efficiency is similar to that of the passive bus system due to the same message and word formats. However, both the message retries avoiding scheme in the active bus system and the fact that direct collision between messages does not occur in this approach (thus eliminating the colliding time-out of the passive bus) make the active bus system more efficient in the view of data throughput.

Switched Network (Ref 50)

The switched network approach evolved from investigations at General Dynamics/Fort Worth Division. In essence, the technique works very much like the public telephone

system, where physical connections are established between devices only when a message is to be transmitted and only for the duration of the message. The basic hardware to accomplish switched network communications has been demonstrated in the laboratory, while the protocol is currently being implemented in hardware and firmware, and a demonstration is scheduled late this year. The basic topology is as shown in Figure 36 (page 113).

A key constituent of a switched system is the switching element (center). Each is implemented with a number of individual ports, one for each attached intelligent terminal. Switching commands are provided by the terminals. A switching center port may be connected to a terminal or to a port on another switching center. In either case, it responds to commands directing it to tie its lines to those of another port in the same center. By external wiring, the same switching center port can therefore support a terminal directly or act as a gateway for a long distance connection between switching centers.

The architecture is considered to have ability to use fiber optic media if necessary, avoiding the splitting of energy problems by the point-to-point communications provided. In addition, each port is connected to its terminal with a full duplex line, which allows each connection to be tested before data is transmitted.

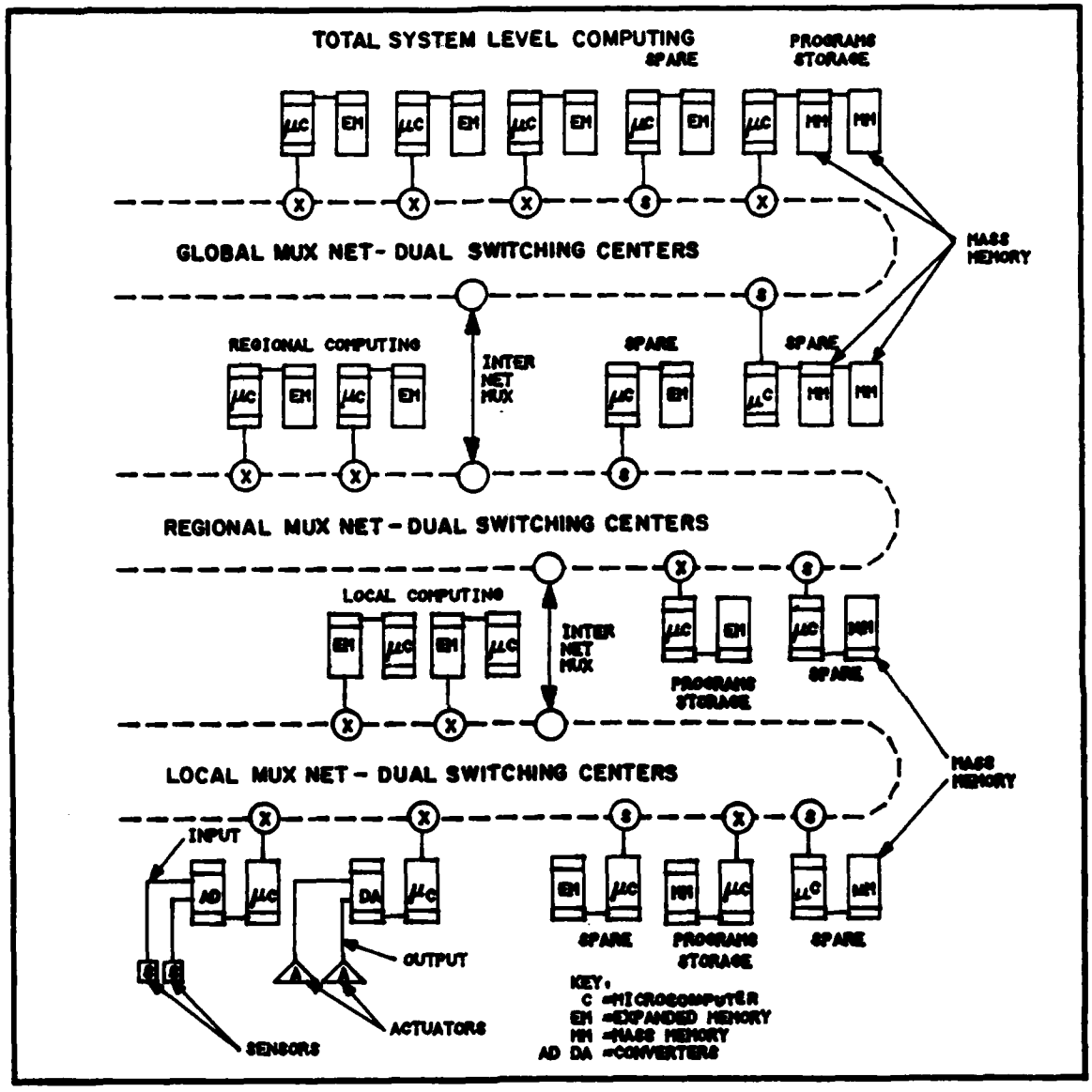


Figure 36. Switching Network Architecture (Ref 50)

The technique assumes the use of "smart" terminals (controllers) which have direct memory access to the memory of the main computer and which can act as both a passive terminal or active controller. Only one terminal is needed

per processing element to provide communications with any other element in the system, anywhere in the hierarchy, but systems which have a great deal of communications would be normally placed on the same switching center.

Any terminal wanting an information transfer first issues switching command to establish the link, and then transmits data as indicated in Figure 37. A message identification word always precedes each message to identify the message content. Parity is transmitted with each word.

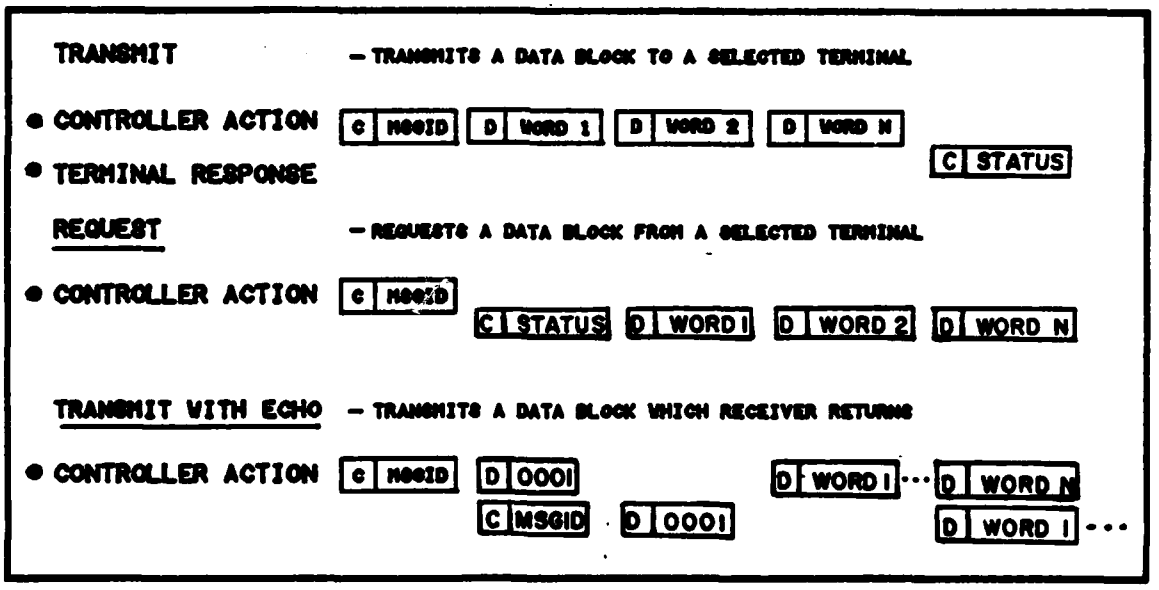


Figure 37. Switched Network Message Types (Ref 50)

Initiation of messages is controlled by command tables in the memory of the processor initiating the transaction, with a series of consecutive commands constituting a command

table. Each terminal, once initiated by its associated computer, will process all the commands in the command table before requiring further support from the computer. Figure 38 shows the format of each command in the command table.

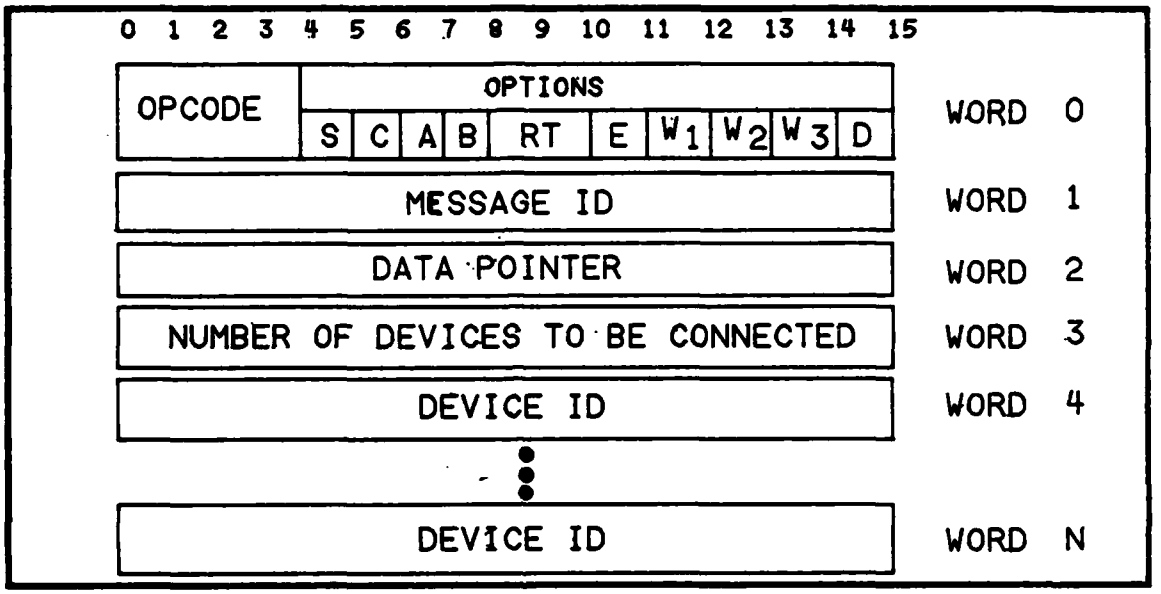


Figure 38. Switched Network Command Table Format (Ref 50)

The first word (word 0) indicates whether the command is a transmit, a receive, a "broadcast" call etc. (including video and video broadcast commands), causing the switching centers to switch video as well as digital lines. Maintaining the connection to a subsystem after the completion of a message is also provided as a command by this word, as well as a Selectable Wait specifying how long to wait for a busy terminal to become free before aborting the message.

The third word is a pointer to the location of the ac-

tual data to be sent, while the fourth and subsequent words in each command table are used as pointers to routing tables, which contain information used at the switching level in completing the desired connection(s).

The protocol requires each switched network terminal to support only one command table at a time, but other messages may be received simultaneously on the second switched network channel.

The approach is highly modular in nature, so that systems can be constructed from standard modules. This is the main advantage of the technique, because the modular structure may enhance the system stability, improve interbus communication, and eliminate the lack of easy growth capability found in some classical bus oriented systems. On the other hand, the considerably more hardware and wiring required can decrease overall system integrity and increase total size and weight of the systems.

The approach supports block and string message transfers, as well as trigger message transactions. The message latency could be decreased by placing subsystems with high ratio of communications to the same switching centers or to the centers that are close enough. However, the overhead imposed by the protocol naming all centers to be switched and searching for available paths, can make this latency unacceptable if longer paths are necessary for any reason.

Bus efficiency, as defined for the purpose of this study, does not have the same meaning for the switched network approach as has for all other techniques described in this chapter. Namely, for the same amount of data transferred, bus efficiency in the switched network considerably depends on the overhead imposed by naming of all switching centers involved in the transaction. However, to have at least a rough feeling about the bus efficiency, it is calculated to be 25 percent for only one data word transferred through a center where both the source and receiver are connected.

There was no data available about the signal modulation code and synchronization pattern used. From the fact that no clock lines are shown in Figure 36, p.113, it can be concluded that one of the self-clocking modulation schemes is used.

The use of the mass memory approach can be critical in the case of heavy load in the network.

The technique is still under development, so that data about software complexity was not available when this report was written. The software can be expected to be fairly complex, as much as interface software in any switched matrix technology.

DP/M--Distributed Processor/Memory System (Ref 49:68-76)

DP/M system was designed for the computational requirements of air-to-ground attack and electronic warfare mis-

sions on an advanced aircraft. The initial design was done by the Honeywell, but revision and more detailed design was done by the Texas Instruments.

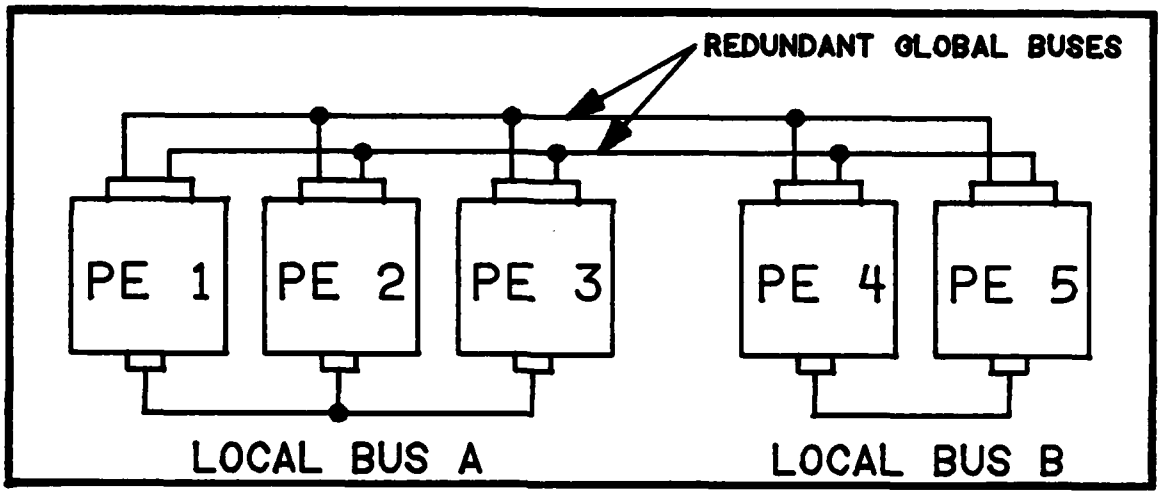


Figure 39. Texas Instruments DP/M Configuration (*Ref 49:68*)

The structure consists of a single (Honeywell) or dual redundant (Texas Instruments) global bus connecting all of several local buses and processors (Figures 39 through 41).

The global buses carry low rate and system-wide messages, while local buses provide the higher data rate paths. The local buses are switched in the Honeywell system and fixed in the TI approach.

The total data rate requirements were about 500 Kbps to 760 Kbps, and speed was one Mbps raw bit rate. The medium is twisted shielded pair wire with maximum length of 300 feet.

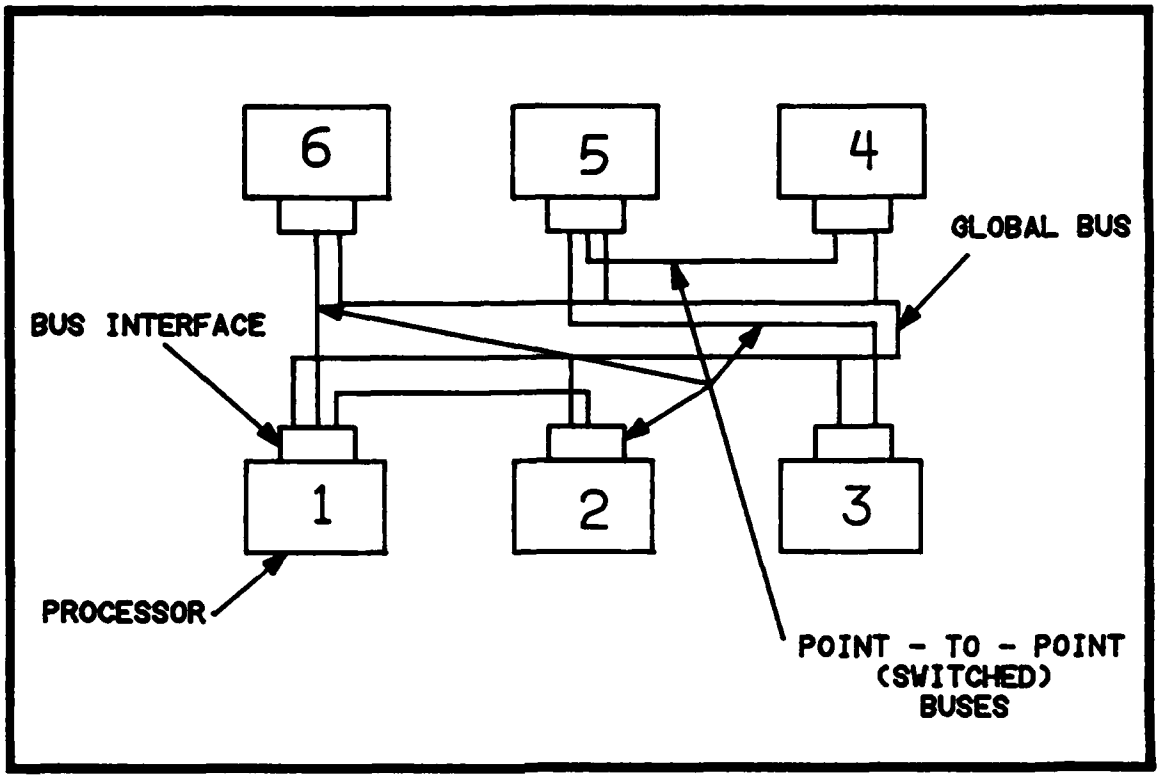


Figure 40. Honeywell D/M Hybrid Bus Topology (Ref 49:69)

Modulation is Manchester-II and synchronization is by message. In the TI technique a "header sync" follows the header word in each message, delimiting the header from the message text.

One-to-one and broadcast transactions are supported (only broadcast in the TI solution). Following the last data word is a bus sync indicating the end of message and reallocation time. There is no response from the receivers.

Message lengths are from zero to 512 data words in the Honeywell, and up to eight data words in the TI approach.

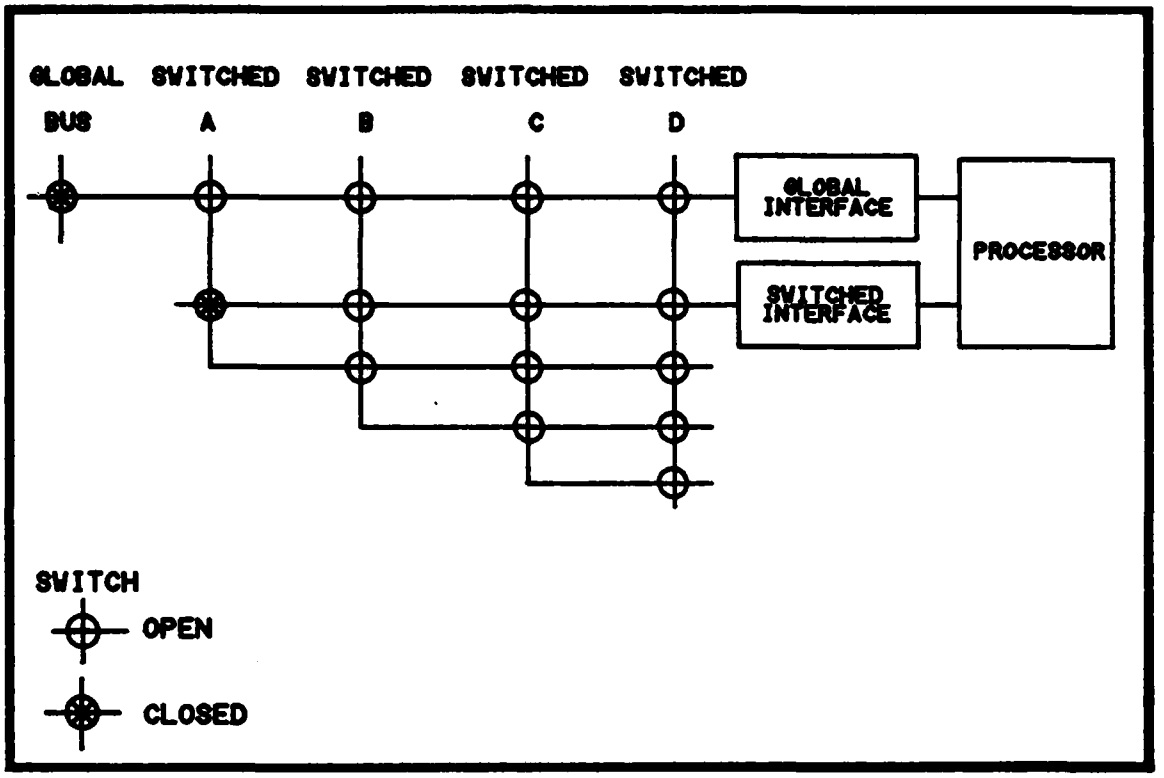


Figure 41. Honeywell DP/M Switched Local Buses (Ref 49:69)

In the Honeywell solution there is no header; the message begins with a variable length name which is recognized by four software-controlled registers in the BIU. In the TI solution, a 1024-bit table in the memory of the processing elements serves as an indicator which message ID numbers are to be accepted.

The global and local buses are controlled and allocated identically. The local bus has its own control hardware, separate but equal to the global bus hardware. The bus is reallocated at message length-dependent intervals. The Honey-

well explores their Vector-Driven Proportional Access Mechanism (Figure 29,p.94), while TI utilizes a prioritized round robin scheme: each processing element gets control of the bus once in every "n" time slots, where "n" may differ for some processing elements, allowing prioritized allocations.

In fact, both the control schemes are implemented in a similar manner, using implicit polling techniques. The BIUs "know" their turn by counting bus allocation time slots (in the TI approach) or by the table lookup algorithm with pointer increments at each EOT (in the Honeywell approach). The primary advantage of these distributed control approaches is that control signals are decoupled from data transfer paths. The primary disadvantages have been already discussed: implicit polling technique requests that subsystems data transfer rates are known and set in advance, thus introducing inflexibility when subsequent changes of the system are needed. In addition, problems arise when a BIU overlooks its turn (or sees a nonexistent EOT, which is not likely but is still possible). As has been discussed (pages 95-97), such problems require special handling, thus complicating software and decreasing system integrity to some extent.

The system integrity is further lowered by the fact that there is no response from the receiver(s). The acknowledgments are essential for system integrity and the message

sender should know what happened to the message during the transfer (at least for some classes of messages). The situation is even more aggravated by the Honeywell solution for receivers that react on any invalid message by simply ignoring it. The TI terminals ignore messages with invalid headers, while garbled data can be used under some circumstances.

The local buses are not backup for the case of a global bus failure. Only one global bus can be active at a time (in the TI system there are two redundant global buses, while Honeywell's switched busses are backups for the global bus). The backup bus switchover procedure is also decentralized and resident in each processor's memory.

There was not enough data to calculate bus efficiency for either system, but considerably longer messages permitted in the Honeywell solution lead to the assumption of its higher bus efficiency percentage.

Transparent Contention System (Ref 53, 55, and 61)

The Air Force Wright Aeronautical Laboratories (AFWAL) at the Wright-Patterson AFB, Dayton, Ohio, has recently demonstrated the Continuously Reconfiguring Multi-Microprocessor Flight Control System (CRMmFCS). The system utilizes originally developed concepts of "transparent contention", "virtual memory", and "volunteering for tasks" in a contin-

ued research program.

The essential elements of the bus architecture are shown in Figure 42.

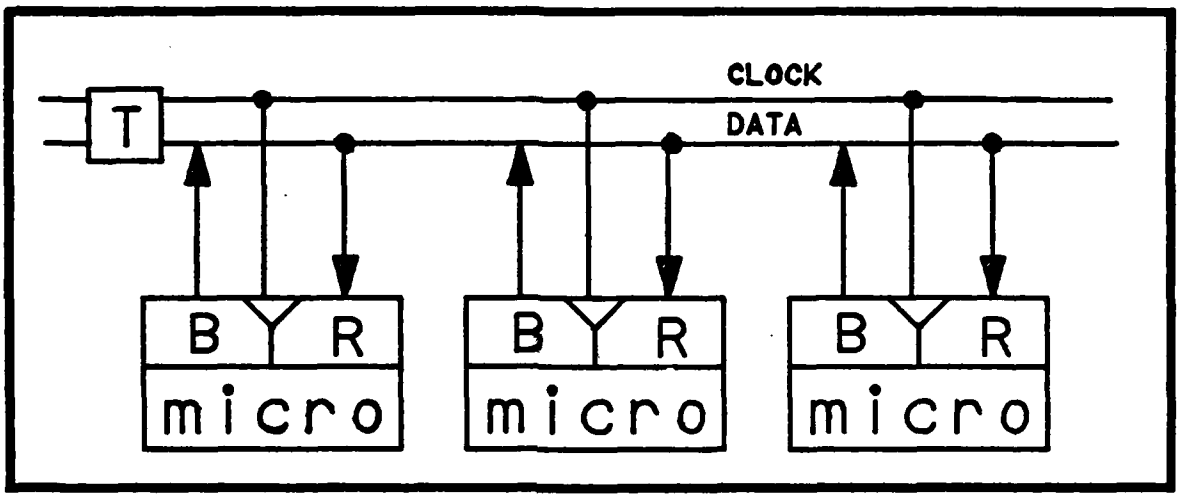


Figure 42. Transparent Contention Conceptual Topology (Ref 55: 30)

The processing elements are interconnected by a common serial bus made up of a data line and a clock line. Each processing module consists of a microcomputer with a broadcaster (B) and a receiver (R). The box labeled "T" in Figure 42 is a bus termination circuit which generates the clock signal and terminates the bus properly.

While the use of fiber optic links has been considered, the actual medium in the laboratory model is wired-OR. The modulation technique is unipolar NRZ code. Synchronization is by a separate clock signal. The bus allocation signal is 9 consecutive one's, transmitted each major frame. The data

word is 37 bits long and starts with a 5-bit sender ID, followed by 11-bit variable name, then followed by 16 bits of data. The data word is divided into four 8-bit bytes, separated by zero's to avoid confusion with the bus allocation signal. All transactions are broadcast only.

The processor wishing to transmit places its information in its local broadcaster FIFO buffer. If the bus is available, the message is transferred starting from the next clock pulse, and any other processor is prevented (using a logical "busy" signal) from initiating a broadcast and overlapping the message being transmitted.

Transparent Contention Concept. The problem of contention, which occurs when more than one transmitter attempts to broadcast starting at exactly the same clock pulse, is resolved by a transmitter-bus interface circuit (Figure 43).

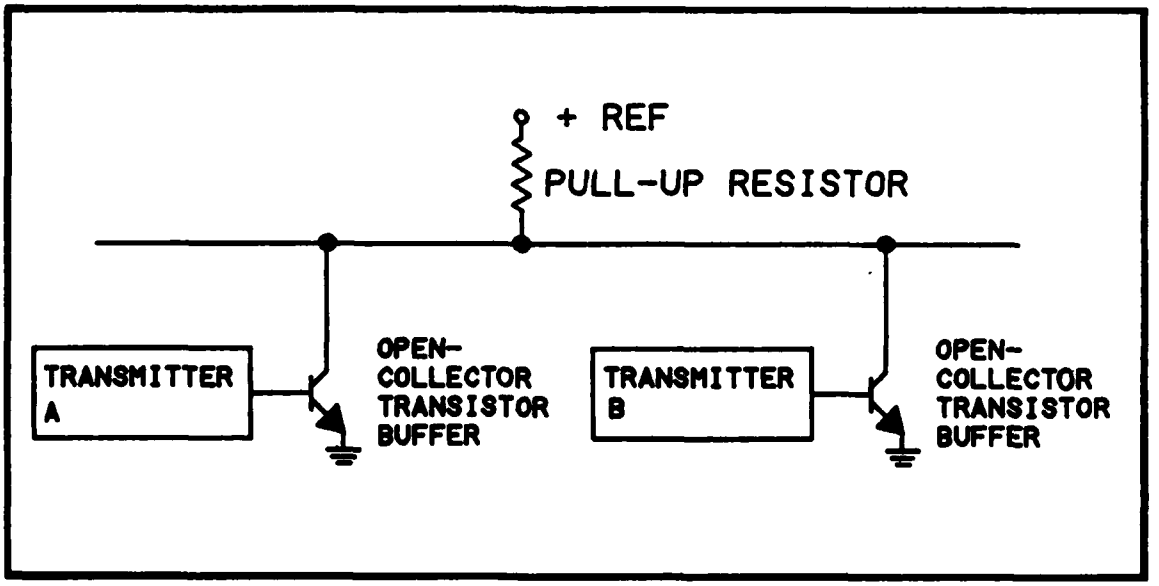


Figure 43. Transmitter-Bus Interface (Ref 55:32)

When any transmitter sends a "zero", it turns on its output transistor shorting the bus to ground. To transmit a "one" the transistor is turned off, allowing the bus to be pulled high by the pull-up resistor if there is no other transmitter(s) shorting the bus to ground at the same time.

Every transmitter constantly compares what it is trying to put on the bus with what is actually there. In the event of disagreement, the transmitter simply stops sending and wait for the bus to become available again. The disagreement is noticed only by the transmitter(s) trying to put a "one" on a bus. Those puting "zero" on the bus are not aware of any competition. The net result is that logic zeros have an inherent priority on the bus, as shown in Figure 44.

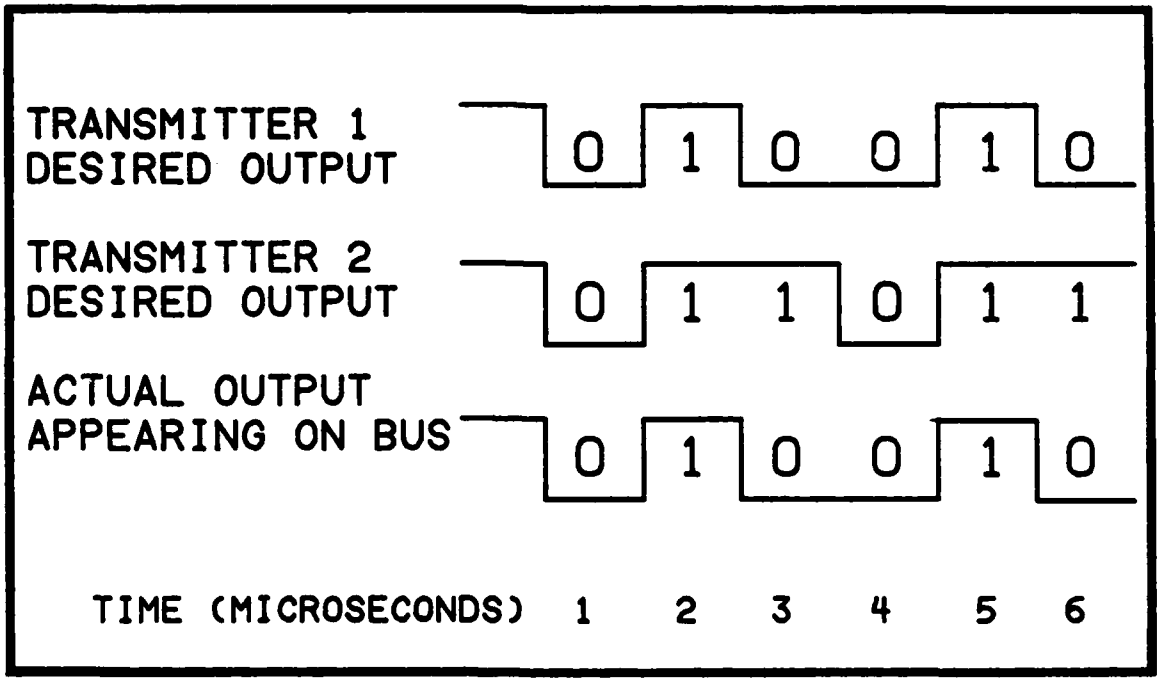


Figure 44. Bus Contention Arbitration (Ref 55:34)

The competition illustrated at the third interval shown in Figure 44 will actually cause transmitter 2 to stop transmitting because it is only one that noticed disagreement. As shown in the Figure, at the end of the competition only one transmitter "wins", completing its message. The concept works equally well for any number of transmitters in contention.

Figure 45 shows the transmitter interface of a single processor in a system with four buses.

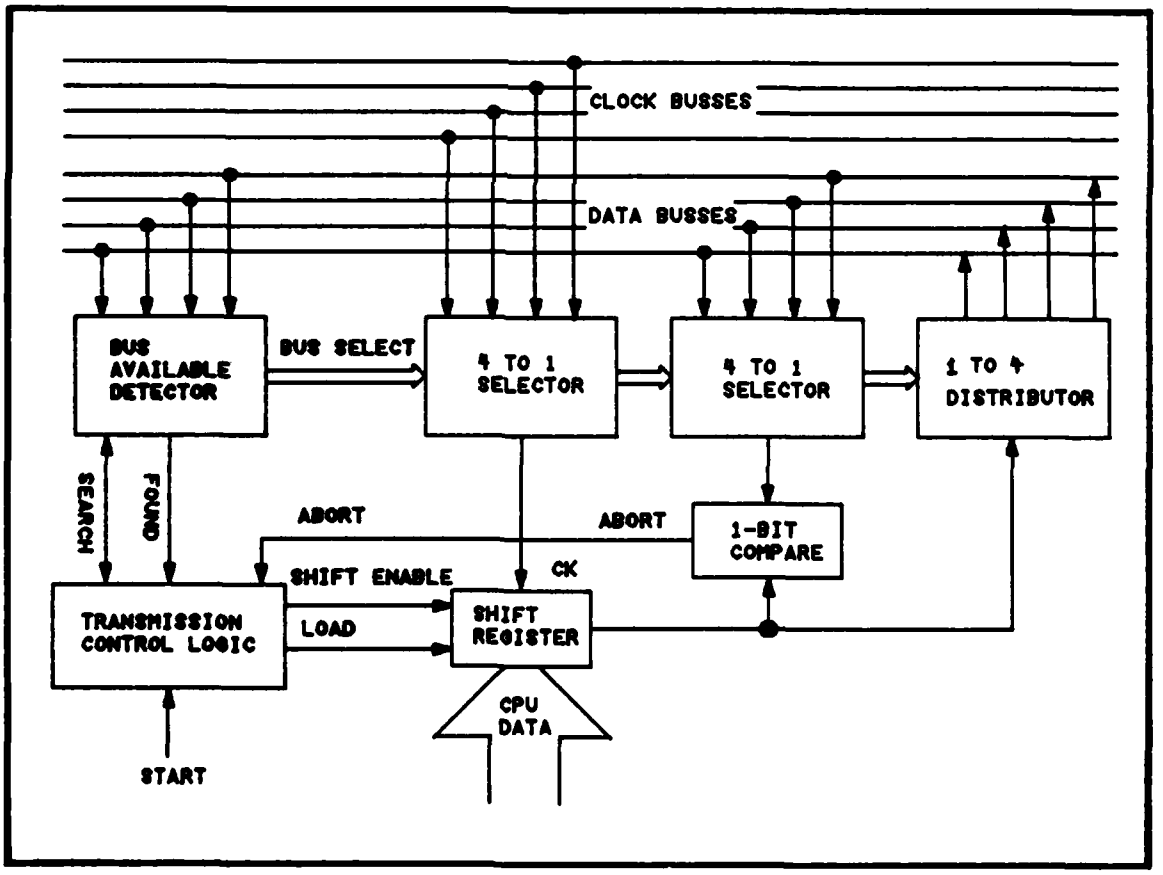


Figure 45. Transmitter Circuit (Ref 55:37)

In the laboratory environment, six such processing modules were connected to the four buses to demonstrate feasibility of the concept.

The circuit is controlled by the "transmission control logic". Upon receiving a "START" signal (from the CPU), this logic instructs the "bus finder" to search for a free bus. When it finds one, it locks two data selectors and a data distributor onto the bus (using its "bus select" lines) and signals the control logic that the bus has been found. The control logic then shifts data out through the distributor. A one-bit comparator monitors for possible disagreements, when an "abort" signal is generated, meaning that the competition has been lost and that searching for another bus should be started.

A two-page buffer (P1 and P2, in Figure 46) is memory-mapped to the local CPU. These pages alternate functions, so that while one of them is being loaded by the CPU, the other is being unloaded onto the bus.

Virtual Memory Technique. Within the laboratory program, an another concept is also implemented. The technique is illustrated in Figures 46 and 47.

In the so-called "virtual memory" design of Figure 46 each processor is given its own copy of the "State Information Memory" (SIM), containing all information about all global variables in the system. In that way each processor

broadcasts data as soon as the data have been computed, such that all variables are updated in the SIM of each processor. On the other hand, any processor reads data from its own updated SIM, without wasting time waiting for the data to be sent if needed. In fact, each processor behaves as if it had access to a common memory.

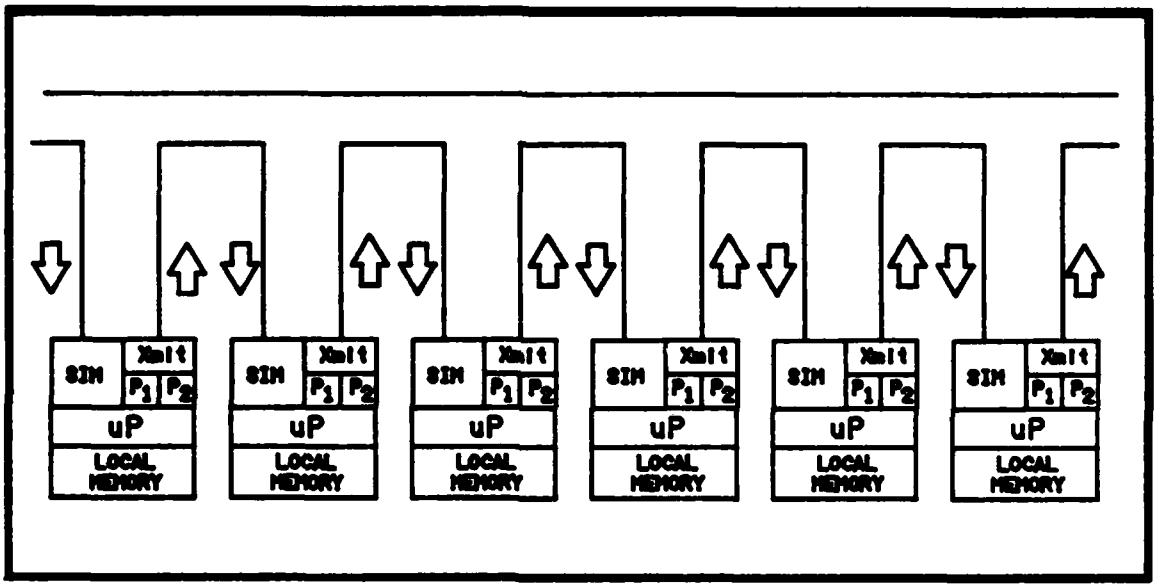


Figure 46. Implementation of Virtual Memory (Ref 55:44)

While the transmitter (XMIT) is explained by reference to Figure 45 (page 126), the receiver and its relationship to the SIM is shown in Figure 47.

The receiver consists of a serial to parallel converting shift register (SIPO) and a RAM which is marked as SIM in the figure. The SIM is mapped into the microcomputer's address space as a block of read only memory, but is acces-

sed by the SIPO outputs as a block of "write only" memory.

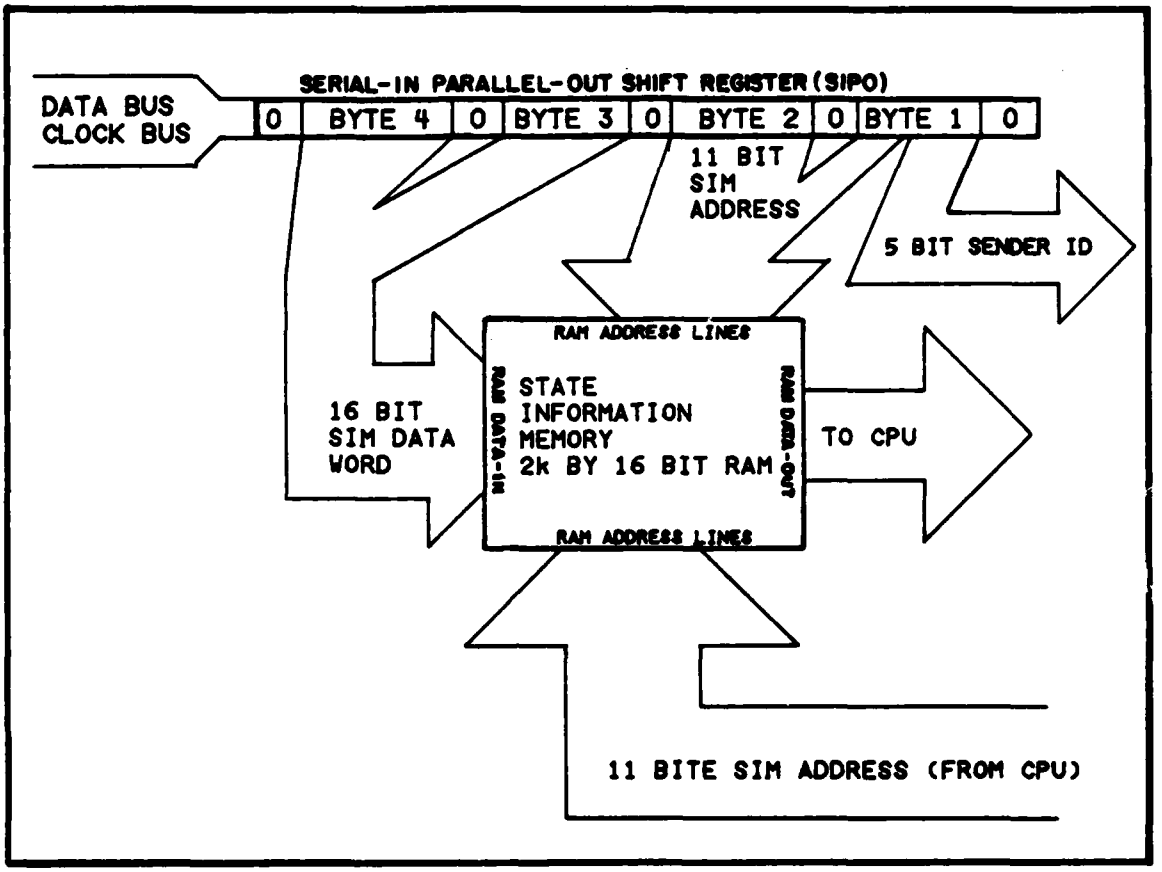


Figure 47. Receiver Block Diagram (Ref 55:46)

Figure 47 is expected to be self-explanatory one, but note that there are "n" SIPO shift registers (where "n" is number of buses) per processing module.

Current high speed RAMs can handle as many as four accesses per microsecond. Since each word appears in the SIPO, in the worst case each 46 microseconds (9 bits of bus allocation time signal plus the 37-bit word), thus the only

limitation are technologically feasible connections of up to 8 buses to the SIM.

Software Partitioning. The virtual memory concept requires transmissions to be scheduled carefully. The scheduling technique is based on the known concept of "quantized software". Namely, time on the bus is divided into equal time slots that are one millisecond in duration for this system. Having messages of 46 bits in length with each bit taking one microsecond, it is possible for the system to transmit 1000 ÷ 46 = 21 complete messages over a bus during each time slot. Thus, the system's four buses are able to transfer up to 84 messages during each time slot. The transparent contention process packs all messages one after another, distributing load equally between the four buses. When all messages are transferred, the buses sit idle until the next time slot starts.

The system's software is modularized in chunks of up to one millisecond in duration (so-called "millimodules") and at the beginning of a time slot the variables being processed during the preceding slot are ready for transfer. This software partitioning is based on the execution times, rather than on functions performed. Because of the difficulties to adjust processing chunks to exactly one millisecond frames, some "millimodules" are shorter than one millisecond. It clearly results in wasting time and in an additional,

although not significant increase in message latency, but note that the method allows exceptional efficiency in task scheduling. Namely, it is now possible to know exactly in millisecond when a variable will be ready for the use by other software tasks and when the tasks can be scheduled. In most applications, such feature can make software design and testing a lot easier and is a well justified trade-off for some greater message latency.

The six-processors, four-buses laboratory implementation is designed with only 22 out of 84 possible messages scheduled for transfer during each time slot. In that way, the system integrity is enhanced because one half of the existing four buses can fail, but the system will still be operational, without any reconfiguration procedures requried.

Although the authors claim 100 percent bus utilization and efficiency achieved, the actual bus utilization and bus efficiency (as defined for this report) are only 25.3 and 34.8 percent, respectively. This is a consequence of the millimodule partitioning method and significant overhead in each of the 37-bit word messages plus the nine bits of the bus allocation signal. Note, hwoever, that the bus efficiency is better than efficiency of any other approach described in this chapter, when compared on one data word message basis.

Message latency is less than one millisecond.

Task Volunteering. In such a system, there are always more processors than global tasks to be performed, due to the redundancy requirements. This fact led to another major outgrowth of this research program: the development of the autonomous control method for dynamically distributing tasks among a group of processors without using a central controller. The method is defined as "a scheme whereby each processor independently determines its own next task based upon the current state of the system" (Ref 55:68). Software modularization, as discussed earlier in this section, allows that all tasks are precisely scheduled and documented in the so-called "task assignment charts". The same chart exists in each processor's memory and the processors know which tasks should be executed within any time frame. A process, called "processor volunteering", is used to distribute tasks between processors during every reconfiguration cycle. Each processor performs a pre-programmed self-testing procedure, and if it finds itself healthy, will "volunteer" for a new task by placing a "1" in the volunteer status table at the place corresponding to its identification number. Otherwise, the processor is assumed to have not volunteered.

A healthy processor will examine the table and assume that the processors with smaller ID numbers and which have volunteered will take tasks in the order that the tasks are numbered. Thus, the processor takes the next task for itself.

As an example: processor #5 will take the task #5 if all of processors #1 through #4 volunteered, but if two of them did not, it will take the task #3 and so on). Tasks are ordered either in a priority order (when task counting starts always from the first task), or are looped and the starting point is random but known to all processors (in the laboratory implementation the value of a certain variable with high changing rate is used as the marker to the starting point). In that way, reconfiguring the system from time to time is a normal procedure, rather than an emergency.

The net result is that faulty processors are out of the system almost immediately after failures occur. Furthermore, even in the case that all processors are "healthy", due to the randomized starting point a processor performs different tasks after each reconfiguration cycle. This enhances the system reliability. Namely, if a processor is for any reason unable to recognize that it has failed and volunteers, it will probably perform some different task than the task of the previous cycle. Thus, errors will not accumulate in the value of only one variable, because the value of erroneous variable will be refreshed by a healthy processor within the next time frame. In addition, other processors can shut down the faulty one if there exists a consensus about which processor was volunteering through several cycles although it was not allowed.

There are indisputable benefits that can be gained from the approach and this is the reason that more attention is paid in this chapter to the transparent contention technique than to any other. It seems, namely, that the approach has been successfully integrated in a system that gathers most of its advantages from inherent characteristics of a fully distributed technology.

However, some drawbacks are also recognizable:

(1) The need for the clock wires and bus termination circuits introduces more hardware, thus more weight and less reliability for the bus itself;

(2) Due to transistor leakage current, open-collector transistor direct connection to the bus limits the number of the subsystems which can be connected to one bus. Also, it introduces a possibility that any shorted bus-buffers may "kill" the bus. The buffers are considered as part of the bus, so that failure of a buffer is the same as failure of the bus, which clearly means many more single points that decrease the bus reliability;

(3) The effect of propagation delays may limit line length to the extent that it prevents implementation of the concept at the global system level;

(4) And finally (but perhaps most importantly), a sender never knows what has really been received and stored in the receivers memories (SIMs). Transient errors in a noisy

aircraft environment may cause an error propagation over the entire system: a bad data stored in the SIM at only one receiver can be the cause of miscalculation of several variables to be transmitted and then received in all other SIMs, causing several new sources with bad data and the error propagates. The solution proposed by the system designers is triplication of each variable in the SIM and a variable must be "voted" before it can be used for the next stage of a calculation. It introduces more software complexity and can also mean a greater message latency.

Frequency Multiplex Systems (Ref 22:63-73)

The use of frequency multiplexing techniques to distribute information over coaxial cables offers an advantage over time multiplexing. The technique is compatible with all signal types and simultaneously provides continuous outputs to all subsystems.

Frequency multiplexing techniques are widely used in cable television systems; however, no single frequency multiplex bus configuration with general applicability to avionics equipment has been identified. Namely, there are a number of constraints which affect the utilization of frequency multiplexing for transmission of signals in aircraft.

An ideal frequency multiplex bus for avionics would allow direct exchange of information between any of the

subsystems on the bus regardless of the equipment position on the bus, accommodating multiple channels over a broad bandwidth.

This implies the use of directional couplers to inject-extract useful information onto/from the bus, and the utilization of amplifiers (which are nonreciprocal devices) in order to overcome the losses due to the couplers and the lossy transmission line, introducing significant limitations on the overall capability of the configuration by the position of the directional devices on the bus (Figure 48).

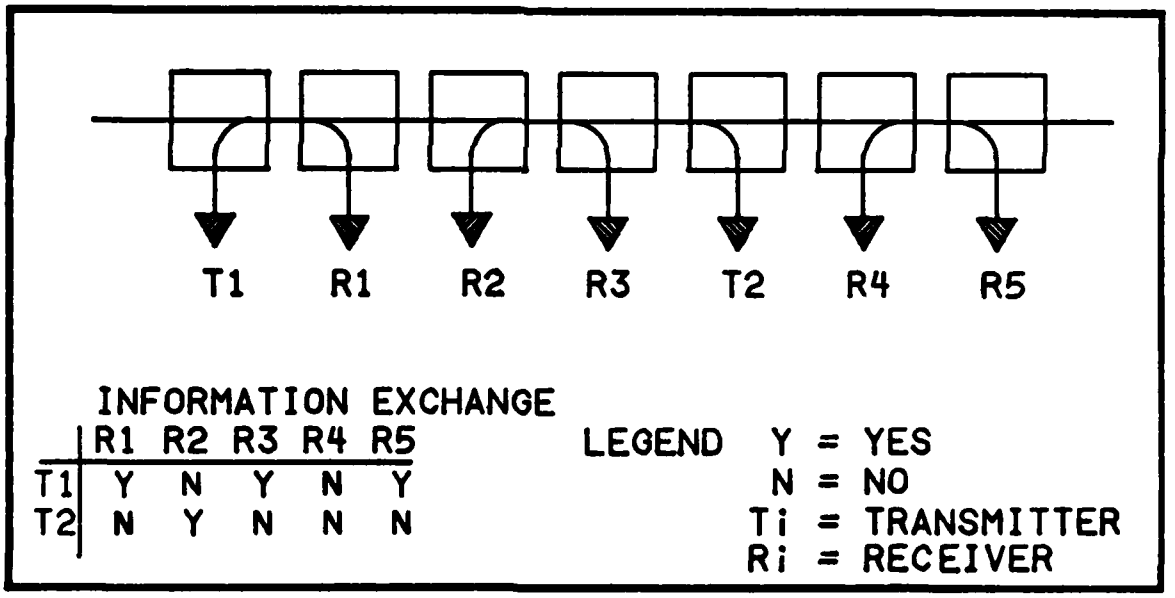


Figure 48. Directional Frequency Multiplex Bus--Position Dependent Configuration (Ref 22:65)

A transmitter can send information to any number of receivers downstream, but a receiver upstream from the trans-

mitter cannot receive the signal, limiting the ability to exchange information by the initial configuration of the bus (the use of in-line amplifiers in the position dependent configuration is limited and once incorporated further reduces bus flexibility).

Commercial systems (Figure 49) overcome the configuration limitations by having all the transmitters send signals upstream to the "head end" where they are converted to a different channel and retransmitted downstream for receipt; the requirements for in-line signal amplification are met by frequency dependent bidirectional amplifiers (low and high pass amplifiers in opposite directions).

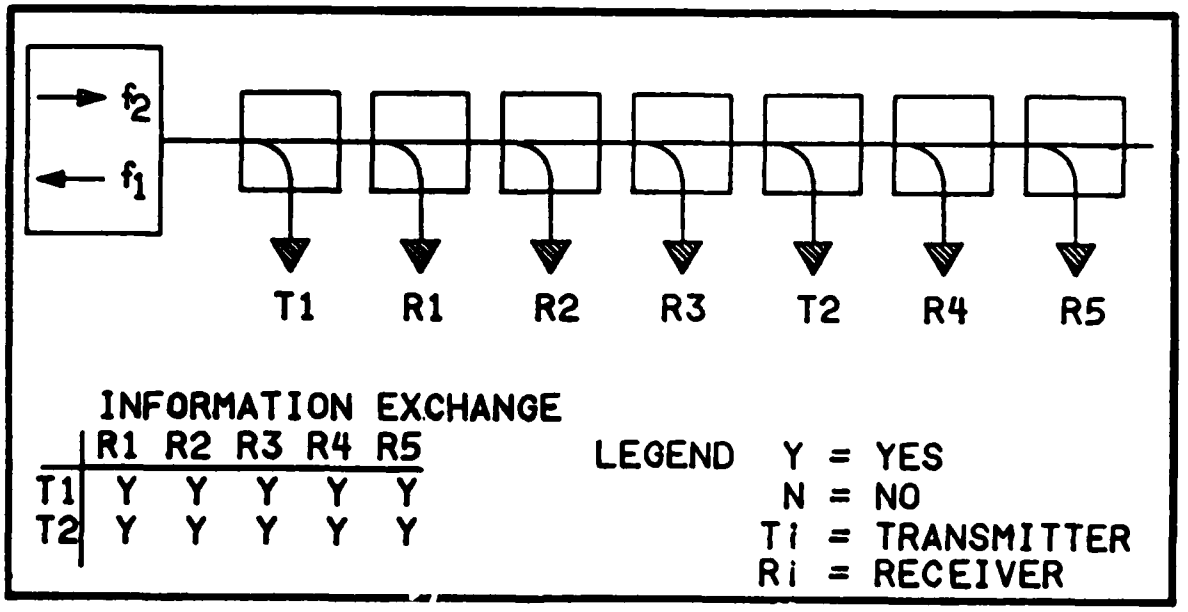


Figure 49. Directional Frequency Multiplex Bus--Head End Configuration (Ref 22:65)

Thus, the bus is in reality two buses operating in different directions and frequency bands on one cable with centralized control.

Neither the position dependent nor the "head end" bus configuration is applicable to future avionics installation. The position dependent configuration does not offer growth capability and may require excessive cabling to attach the subsystems to their correct locations on the bus. The "head end" approach requires an excessive hardware investment, occupying the two frequency channels for each single transmission.

Dual direction coupler configuration capable of injecting or extracting signals on/from the bus in both directions simultaneously is realizable, but losses would be approximately 6dB greater than for a single directional coupler (Ref 22:64). This further limits bus length if in-line amplifiers are not to be implemented.

Figure 50 illustrates a configuration proposed by the NADC, which could be applicable. In such two-cable configuration, in-line amplification is feasible and the location of the amplifiers will not limit the bus operation. However, the couplers appear complex, although they exhibit 3dB (as opposes to 6dB) additional insertion/extraction loss for a given value through insertion loss (Ref 22:66).

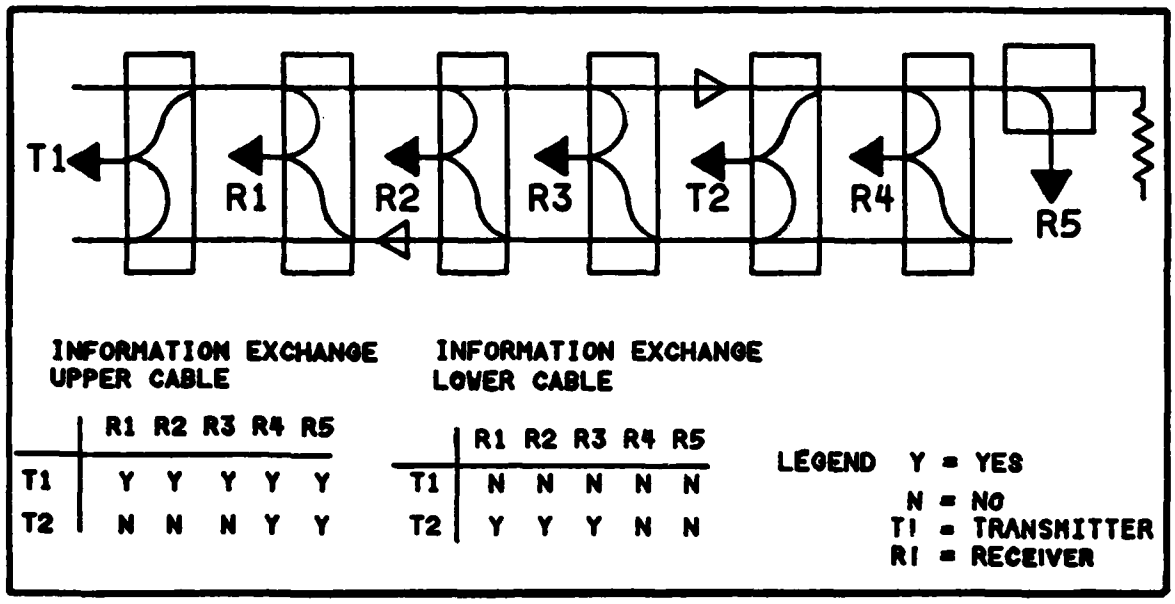


Figure 50. Two-Cable Frequency Multiplex Bus (Ref 22:67)

The development for such implementation would include development of bus frequency conversion modules, bus couplers (transmit, receive, and transmit/receive), in-line amplifiers, and filter assemblies. There was no available evidence that such development has been initiated, and the technique is presented in this report only in order to show a possible alternative to time division multiplex systems with some of the advantages of frequency multiplexing techniques. Some disadvantages and problems to be solved are presented, as well.

This is, probably, the right place to mention that there is relatively new technique which can be considered

within the class of frequency multiplex systems. Namely, the optical wavelength division multiplexing (WDM) systems, with signals transmitted on different wavelengths through a single fiber, can have increased information capacity and fault isolation properties over single wavelength optical systems. The technique is not more elaborated in this report due to several technological problems that still remain unsolved at present, but the interested reader may refer to Ref 82.

ARINC 575 Digital Air Data Multiplex System (Ref 41:48-51)

Digital data buses are utilized in today's commercial aircraft, but only on an individual system basis. Wider usage is anticipated in the next generation (Ref 26).

The current commercial standard (the ARINC-575 specification) was written in 1971 by Aeronautical Radio, Inc. (ARINC), sponsoring the Airline Electronic Engineering Committee (AEEC) that formulates standards in the form of characteristics and specifications for airline electronic equipment and systems.

The ARINC 575 bus operates only in a broadcast mode. A single transmitter sends data at a 11 Kbits/sec rate simultaneously to a maximum of 20 receivers connected to the same bus. Bipolar RZ modulation is employed. There is only one transmitter on each single bus and if a receiver needs data from more than one source, it has to have a separate BIU

connected to the separate buses. There is no acknowledgment sent by the receiver to the sender. Any garbled messages will be lost since the receivers cannot request a repeat, nor can the transmitter ever know what has been received by the receivers.

The message contains a header that names the variable being sent, followed by the data and any sign or status indications. A message is always one 32-bit word.

The bus is transmitter controlled. There is no possibility for any contentions to occur on any bus since only one transmitter transmits on each single bus, thus bus allocation signals are not needed. Another advantage of the technique is that the transmitter sends a data word when the word is available to be sent, which is very desirable feature for real-time avionic systems. However, such good message latency characteristics are paid for by oversizing of the system cabling and by multiplicating the BIU's inputs. The problem is further aggravated if redundancy is required.

As seen, more suitable solutions for military aircraft do exist, and this approach is briefly presented only as an illustration of existing status in commercial aircraft.

Chapter Summary

In this chapter, fourteen different approaches to the same problem of data multiplexing in avionic systems are

given. The solutions are described and analyzed according to the multiplex system key issues defined in Chapter III.

The techniques represent the current system design base and the advantages of particular technologies are included in the definition of "optimal" architecture characteristics in the next chapter.

V. Optimal Architecture Definition and Conceptual Design

Having defined the networks' figures of merit in Chapter III, and discussed the advantages and disadvantages of different technologies in Chapter IV, it is now possible to define the characteristics of an "optimal" architecture for digital avionics systems, as follows:

(1) The architecture should reflect the results of the successful current system design base, i.e., should involve as many features of the existing time-division serial multiplex bus techniques as possible to integrate into a single architecture.

(2) The architecture should be simple and general enough to be standardized for use in the entire aerospace industry. The very same architectural philosophy should be utilized from the system's global bus down to the lowest subsystem level. This will allow a massive, cheap acquisition of the basic components that can be integrated in various ways to build a system as required. In this way, the main goal of any standardization will be achieved.

(3) The system designer must be free to utilize the philosophy by using any current technology, according to the particular system requirements. This translates into the technology independence of the "optimal" architecture philosophy, avoiding the trap of technology dependent hardware

standardization that can freeze possibilities for improvements as technology advances.

(4) Multilevel, hierarchical, lossy configuration of system topology provides possibilities for functional partitioning, both in hardware and software. This will allow desirable flexibility in multi-mission applications of the same system, including the possibility for subsequent growth of the system. The maximum fault isolation between subsystems will also be achieved in such a way.

(5) Distributed bus control with a purely transparent contention bus allocation scheme is required to support the flexibility and integrity features of the system topology.

(6) The system must be independent of physical characteristics of the transmission media.

If wire cables are utilized, the transformer coupling is preferred.

(7) Bus length of up to 500 feet is considered to be enough for any application in the foreseeable future.

(8) Logical limitations imposed by the system protocol and message formats on the bus size (the number of terminals allowed), must not be greater than those imposed by the utilized transfer medium physical limitations. Doubling the number of 32 terminals (this number is common today as the limitation on bus size), seems to be enough for the systems in the years to come.

(9) The architecture is to provide transmissions with up to 10 Mbps data rate. Bus efficiency and message latency should be improved as much as possible by decreasing the message overhead. The information-only transfer philosophy will support these improvements, bringing up the new quality to the bus utilization definition (will be discussed later).

(10) A self-clocking, baseband signal modulation is to be utilized.

(11) Word and message synchronization by unique waveforms in the serial bit stream is to be used.

(12) The 16-bit information field in data words is to be utilized.

(13) The message protocol and possible system applications should be decoupled and mutually independent.

(14) Hierarchical message addressing to the software defined names will support the hierarchical system topology and enhance the features of distributed bus control.

(15) The system is to allow both periodic and aperiodic messages transfer, with aperiodic message being prioritized.

(16) Variable-length block and string messages transfer should be possible.

(17) In addition to standard techniques to decrease software complexity and costs (top-down design, partitioning and modularity, higher-order-language usage, etc.), the system software interface must match the hierarchical, distri-

buted nature of the architecture that allows higher level of "smart" hardware support.

(18) Some sort of dynamic reconfiguration is to be provided to enhance system integrity so that recovery procedures require little or no emergency reconfiguration in the case of bus or terminal failure.

Error detection should be possible by parity per word and by cyclic checksums at the end of longer messages. Redundancy should be optional, rather than mandatory.

(19) Implicit broadcasting allowing transmissions from one source to unknown number of receivers should be utilized as the primary way of communication. At the same time, acknowledgments must be allowed even when multiple receivers are involved, because the acknowledgments are essential for the system integrity.

Comparison with the AE-9B "Future" Bus. Probably, this is the right place to compare the "optimal" characteristics listed above with the basic requirements, given in Table II on the next page, for the "future high-speed data bus" as recently defined by the AE-9B HSDB Subcommittee (Ref 5:54).

It is noticeable that they match in most of their key issues. The variations are due to nuances in the definitions of the essential characteristics, but there are not significant conflicts between the two sets of characteristics. This makes the "optimal" architecture, as defined in this report,

at least a candidate to be implemented in a hot bench and further refined.

TABLE II

Prime Characteristics Required for Future High-Speed Data Bus

Characteristics	Goal
Distributed Bus Control	To achieve maximum fault isolation and ease of system change
Data Rate	Up to 20 Mbps throughput (dependent on technology available)
Baseband System	Possible application to broadband later
Linear Bus	Multidrop, like MIL-STD-1553
Coaxial or Fiber Optic	Desire system to be independent of physical bus medium characteristics
Protocols	Contention, Token Passing (i.e. Polled/Round Robin), etc.
Integrated Data and Audio	Standard intercom (ICS) and interactive audio functions combined on the bus. Bus to handle minimum of 20 stations
Message (Protocol) Format	Bus to handle all types of message transfer and response (broadcast, request, and request with acknowledgment)
Message Length	4K words, maximum 1 word, minimum
Word Length	16 bits
Redundancy	Optional (single bus allowable)
Bus Length	Up to 150 meters
Bus Size	Up to 64 terminals
Error Detection	Short Message - parity per word up to and including 31 words; Long Message - 16-bit CRC for messages of 32 to 4K words

(Modified from Ref 5:54)

The word "optimal" is put into quotation marks throughout this report to point up the fact that what is optimal for some particular applications is not necessarily optimal for all others. In addition, the listed characteristics represent, in a way, an envelope of the best features, that are either taken from the analysis of very different and sometimes mutually exclusive techniques or are just results of some theoretical considerations. The consequence is that all of the so defined characteristics cannot be easily integrated in a single system and some trade-offs are unavoidable. In this way, the word "optimal" looses its primary sense.

In addition, any study on the subject would not be complete without an attempt to synthetize a system based on the conclusions drawn from the study's results, regardless of how difficult integration of the results could be. Also, low level implementation details do affect overall system characteristics to a greater extent than is usually recognized.

These are all the reasons that an architecture, named as the Self-Managing Multiplex System, is conceptually developed in this chapter. The description of the development should also serve as a guide for the multiplex system "synthesis issues" according to the goals defined at the beginning of Chapter III (pp. 14-15).

Self-Managing Multiplex System

Probably, the following preliminary design of the Self-Managing Multiplex System (SMS) contains most of the "optimal" architecture characteristics. Although some new ideas are included, the concept avoids the pitfalls that are often made when designing complex systems (Ref 76:658): (1) being nothing revolutionarily new in the field, the SMS integrates a variety of already implemented solutions, thus making itself an evolutionary development from a successful current system design base. At the same time, the integration of several existing techniques in a new context, based on the analysis performed, has more powerful application possibilities; (2) the concept is not based either on "if a little is good, a lot is better" logic, or on concentration of responsibility only at one point in the design organization. It is rather directed toward covering as much of desired characteristics as the "optimal" architecture is to have. The really new ideas are introduced only when was necessary to resolve conflicting requirements within the set of desired characteristics.

Basic System Requirements. Any design of a data transfer system must start from recognizing and defining the mission requirements to be satisfied by the avionics architecture. For the purpose of this development, a hypothetical aircraft is considered because of the lack of available data

for modern military aircraft.

According to the published data for the A-7D, carrier based attack aircraft (Ref 49:89, 224-261), with its current federated architecture similar to DAIS, the worst case total data transfer loading is calculated as 84,351 bits/second to be exchanged between 31 subsystems. This gives approximately 21 messages of ten 16-bit words with an average update rate of 20 to 25 times per second.

The hypothetical aircraft is assumed to be capable of handling at least 10 times more total data transfer loading (843,510 bits/second), with 10 times more global variables (messages) and double the number of connected subsystems (i.e., 210 and 64, respectively). The average rate of message update is also to be doubled (about 50 times per second).

In order to exploit already discussed benefits, the system is to be designed as a hierarchical, fully decentralized architecture based on the "transparent contention" control scheme for a time-division serial multiplex bus.

Transmission Medium. The system is to be independent of physical bus medium characteristics. The relatively low data transfer loading led to the decision to use twisted-shielded pair as the transmission medium in the discussions that follow. This type of transmission medium is a well-understood, mature technology, that is inexpensive and reliable. In addition, some further benefits from a new concept can be

gained when such medium is used, as will be explained later. However, the reader should be aware that the physical nature of the particular medium chosen causes different effects when this new concept is employed, and this will be pointed out whenever applicable.

Signal Modulation. The Manchester-II (Figure 11, p.30) seems to be the most appropriate choice because of its wide usage in the field. It is a standardized technique, employed in almost every aircraft data multiplexing system today. There are quite a few manufacturers that are able to cope with any hardware solution required when the Manchester - II technique is used.

These facts will allow an inexpensive, large-scale production of the hardware components for the SMS-based systems in the future, protecting the investments into development and existing production lines that already have been made.

Bus Allocation Scheme. The first of several new solutions for the SMS concept is introduced in this subsection. Namely, if the Manchester-II modulation is to be implemented in a "transparent contention" bus allocation scheme, then some solution should be found for the bus interface units to compare their own current transmissions with actual signals on the bus and decide on whether to continue transmitting or not.

Figure 51 depicts the transparent contention technique

implemented in a new way. An idealized bipolar Manchester coding is depicted. Note that the technique will work equally well if the unipolar Manchester is used, meaning that the concept can be implemented with optical links as well (the optical link technology, namely, requires some sort of unipolar coding because of the unipolar nature of optical links: the presence of light or its absence are the only two possible states on the bus).

In order to introduce the concept, assume that all terminals connected on a single bus have the ability to transmit and to receive at the same time. Assume, also, a distinguishable bus allocation signal is somehow generated and is present on the bus. Further assume that each terminal, having some data to be sent starts transmitting at the bus allocation signal and compares on a bit-by-bit basis what it is trying to put on the bus with what is actually there. In such a case, when the Manchester coded signals are generated by separate sources, the bus will be a summation point. As a consequence, the majority of the terminals sending the same bit will not notice any disagreement, since their receivers will sense the bus voltage transition in the desired direction. The terminals sending a different bit will notice the disagreement and will stop transmitting, waiting for the bus to become available again.

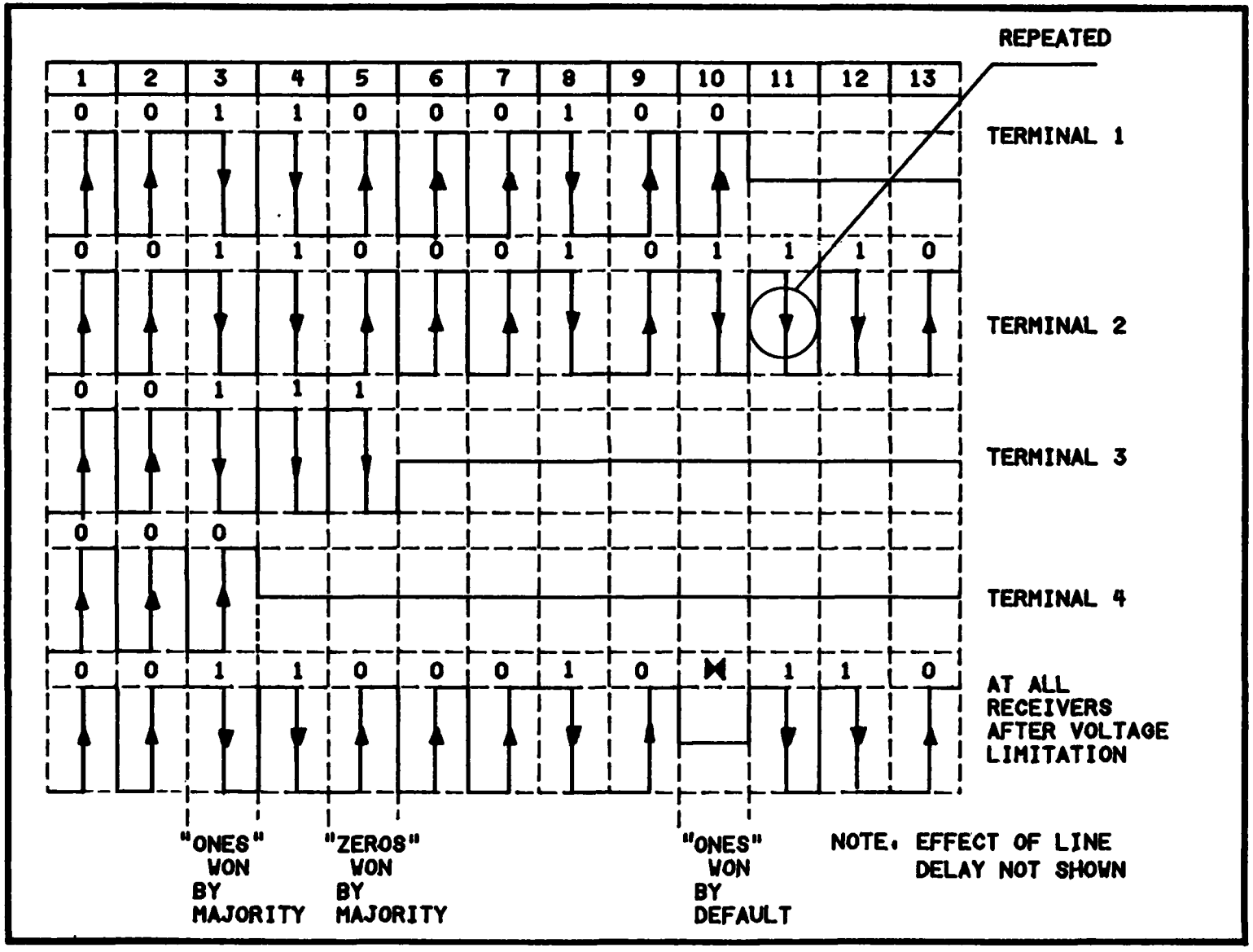


Figure 51. The SMS Transparent Contention Implemented by Manchester-II Coding

Such disagreements occur in Figure 51 at bit #3 and bit #5 frames, thus terminals 4 and 3 stopped transmitting, respectively. The figure shows that in this technique majority rules and there is no inherent priority of one logical state over another. This fact will be very useful when used in the new concept of the self-correcting broadcast/acknowledgments mode as discussed later.

An undefined status occurs in the case when equal number of terminals try to put different bits on the bus at the same time: the net result is zero voltage on the bus, and by default, terminals sending "ones" will win over those sending "zeros". The approach would work equally well if "zeros" won in such a case, but design of the bus interface unit (BIU) is simpler, as shown later, if "one" is given such priority. At this point, there are two possible solutions: either "ones" will be repeated by the sender(s), as shown in Figure 51, or the receivers will automatically replace this "equal contention dead bit" by a "one" (see the modulation correction unit - page 110). In either case the terminals sending "zeros" will stop their transmission.

In the proposed SMS design, the first solution is preferred in spite of the fact that repetition of the bit wastes one bit time. Namely, several successive "equal contentions" may occur and, with the second solution, this would make the BIU design more complicated (precise internal

clock circuits, prevention of automatic restoration when bus is really silent, etc.). Thus, the wasting of one bit time seems to be well-justified trade-off.

Considering the case depicted in Figure 51, terminal #1 will stop transmitting at the bit #10 frame, but terminal #2 will repeat its "one" and complete the message. The message will be received by all receivers attached to the bus, including the receiver of terminal #2. Note that a terminal receives its own message over the bus and then stores it into its state information memory (SIM - see pages 127-129). In such a way, all terminals are "equal" and this allows each terminal continuous self check of its own receiver. This is very important and will be discussed in detail later on.

The number of terminals in competition is not a limiting factor for the concept itself. Only one terminal at a time completes its message without significant message overhead required and without a bus controller of any kind. There is no possibility that two terminals will ever try to send identical words at the same time, as will be shown in the next subsection describing the SMS protocol.

Of course, the idealized Manchester waveforms as depicted in Figure 51 are not feasible. The effects of line propagation dalays and attenuations may be relevant. Different signal rise and fall time, as well as deviations in the zero crossing point that vary from transmitter to transmitter,

must also be taken into consideration. The noise environment of an aircraft introduces additional problems.

Obviously, the on-line summation of such signals will result in rather complex waveforms from which a receiver must extract the bit information. This will be discussed in more details, once the concept is completely introduced.

Message Protocol. The system message protocol is defined in a data-variable addressed fashion. Namely, a message consists of two (or more) 16-bit words plus horizontal parity bit for each word (Figure 52). The very first word in each message contains a 12-bit field for the "names" of variables to be exchanged in the system (i.e., altitude, velocity components, etc.).

Eventual competition for the bus will be resolved based on these names, which are unique for each variable in the system. Thus, there is no possibility that two or more terminals will ever have the right to send data at the same time.

The length of the data words is determined based on the discussion in Chapter III (page 43). In order to have unified word length in the system, the same 16-bit (plus parity) length is also chosen for the first, "contention resolving and variables naming" word.

Word synchronization is employed, rather than synchronization only at the beginning of the message. This allows

better internal synchronization for all the BIUs and is particularly important due to the reliability requirements for the systems employing bit synchronization by extracting a clock from the data stream. This is the case with the SMS technique, where possible complex waveforms on the bus further emphasize such a requirement.

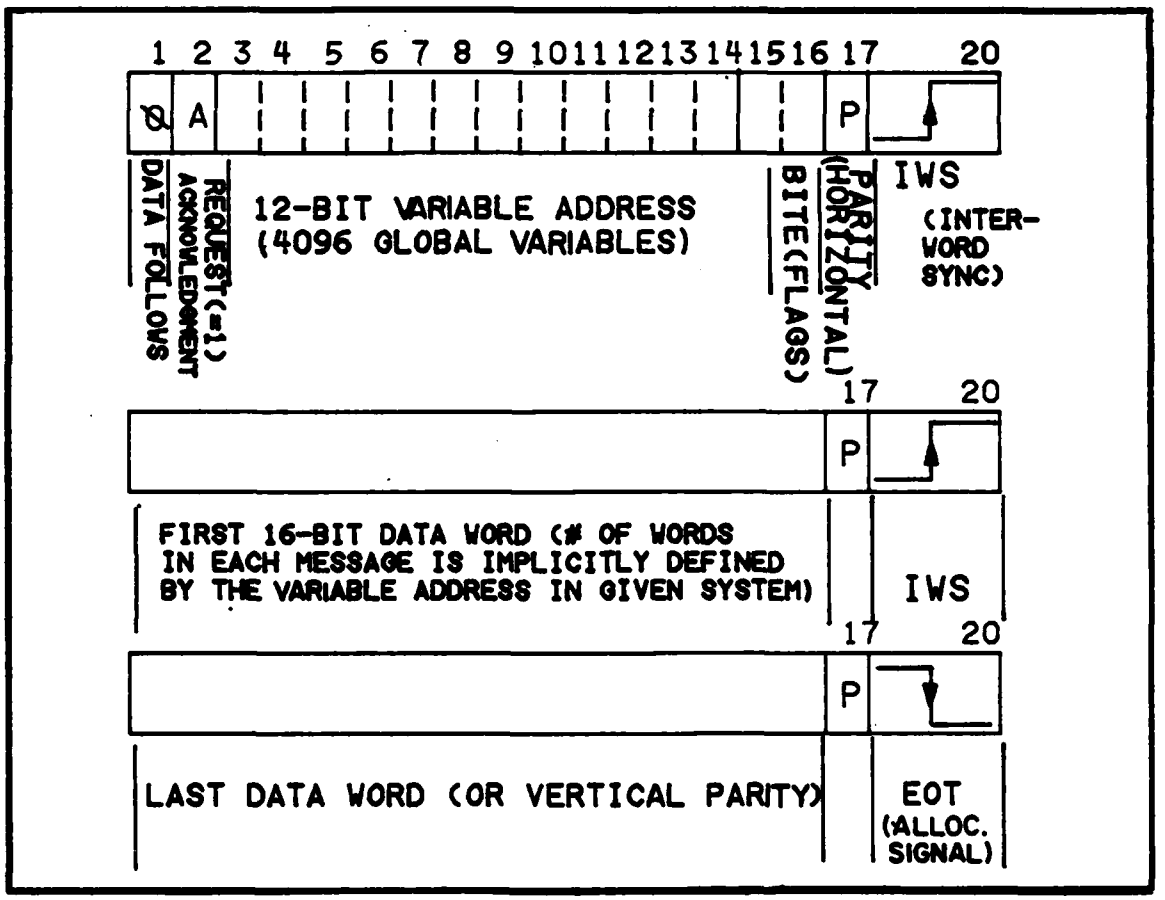


Figure 52. The SMS Message Protocol

The sync patterns are adopted from the MIL-STD-1553 approach, using the data sync (Figure 21,p.74) as inter-word sync (IWS) and the command sync (Figure 20,p.74) at the end of transmission (EOT) as the bus allocation time signal. All data words within a message are separated by the IWS's.

Each terminal has a time-out circuit with a unique time constant, triggering the EOT signal generator within the BIU so that a silent bus will always have an EOT from time to time, allowing the competition mechanism to start. Namely, if there is no terminal wishing to transmit after the normal message's EOT, the bus would be "dead" forever. This is resolved in that the terminal with the shortest time constant will time out first and send a new EOT signal. If it fails to do that, the next terminal will time out and so on. The EOT is sent periodically as long as a new message is not being transmitted. Each EOT signal, sent either as the normal ending of a message or as a separate signal, resets the time-out circuits in all BIUs on the bus. The same mechanism is also used at every initialization when the system is powered up. Time constants are programmable, allowing the system designer to adjust bus allocation signal periods depending on the system requirements.

Refer back to Figure 52. A message containing data for a variable always starts with a "zero" bit in the first word. The second bit is an acknowledgment bit ("one" means that

the acknowledgment is requested and will be used depending on the variable importance. Zero means no acknowledgment is necessary). The 12-bit variable-name address field in bits 3 to 16 allows up to 4096 global variables to be used in the system. The addressing method is intended to be in a hierarchical fashion (see page 46), allowing the system to be partitioned and integrated into a hierarchical topology (see page 17). The next two bits (15 and 16) are used in the built-in-test (BITE) mode as the command signals, followed in the last position (17) by a horizontal parity bit over the first 16 bits. The first word, as already mentioned, is used to resolve any possible competition and is ended by an IWS, followed by as many 16-bit data words as needed (usually one data word per variable is enough). All data words within a message are separated by the IWS's. The last word can be used as a vertical parity word allowing standard self-correcting techniques to be implemented. Thereby, messages are of variable length, and this gives the system designer greater flexibility. A completed message is always ended by the bus allocation time signal (EOT), that replaces the IWS.

The SMS concept assumes only broadcast mode for message transfers, fully utilizing all the benefits of low overhead in such a mode. At the same time, acknowledgments are used in a new and very efficient way to fight transient errors as

will be fully explained later on.

The broadcast mode, combined with the virtual memory technique (the SIM concept, pp. 127-129) offers a very efficient way to handle real-time systems' requirements as pointed out on the referred pages and will not be repeated here.

However, the state information memory technique is now extended in the SMS concept by introducing so-called "book-keeping words" (Figure 53) that are stored in the SIM along with the variables.

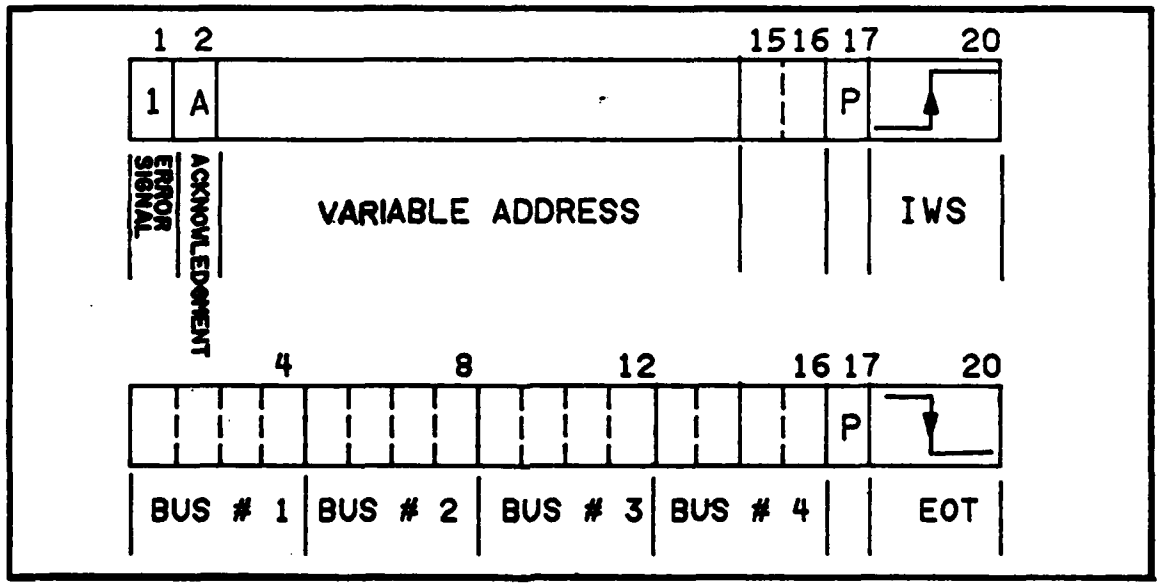


Figure 53. The SMS "Book-Keeping" Word

Namely, in order to reduce the bus load all of the messages will not be broadcast with the acknowledgment request, which means that they are not of critical importance. The acknowledgment mechanism, as will be seen, automatically en-

sures that all terminals (without exception) have received good data. But, if such a possibility is excluded by the system designer for a variable, this means that bad data for such variable can be tolerated in the system if received from time to time (for example, in some non-critical systems such as in-flight monitoring equipment for post-flight analysis purpose). This further means that rejection of such bad messages is not essential even when the error has been recognized by the receiver, and that the sender does not need to be notified immediately.

Having in mind more than one bus in the system due to the reliability requirements, the purpose of "book-keeping word(s)" is to allow different transient errors to occur once on each of the buses, and the sender will be notified only if the same error for the same variable occurred twice on the same bus. In such case the "error signal message" will be sent, competing for the bus in the usual manner. The only difference between data messages and error signal messages is that error messages begin with a "one" in the bit position 1, while content and the number of words describing the errors on the buses are at the discretion of the designer. Figure 53 depicts the "book-keeping word" (error signal message) for a system designed for up to four buses. There are four different errors that can be recorded for a bus: parity, word length, data drop-out, and invalid IWS.

The use of the error signal messages is optional and can be avoided if the system designer wants as small an SIM as possible.

All messages can be acknowledged. This gives tremendous reliability, but sacrifices data throughput, which can be improved by more buses with greater cost, and so on. The trade-off is the choice of the system designer.

It is also possible to design a system that repeats critical data over several buses within the same time frame. Repeating of the same critical message is a technique employed in the Space Shuttle - the receiver's software votes on such messages and uses them only if the software determines that a good message has been received.

On the other hand, the use of error signal messages in the SMS concept replaces a classical bus monitor with as many active monitors as there are terminals on the bus. This can be used as a suitable tool for mutual health-checking among the terminals. It is true that errors are tied to the variables and not explicitly to the terminals, but a variable is generated only by one terminal. Therefore, mutual checking among terminals is involved. Sending of an error signal message is the case when more than one terminal may send the same message at the same time, but it has no importance since only the variable generating terminal will use this message as a trigger for BITE selfchecking procedures.

All other terminals will reset book-keeping words for the variable, suppress the same error message, and start the process all over again.

Bus Interface Unit. This thesis work is intended to be the definition of an "optimal" architecture. Any implementation is beyond the scope of this effort. However, as already pointed out, low-level details usually do have implications throughout the network architecture. This is the reason, along with a need to illustrate the concept, that a proposal for the four-bus BIU implementation is given in Figures 54 and 55. Note, please, that the proposal is given only as a possible way to implement the technique and to show that an actual design would also have to provide full hardware support to raise the system performance and lower the software and maintenance costs.

The concept is illustrated by a BIU conceptual design for four twisted-shielded buses, but the BIU can be connected to optical links as well (the only difference would be the bus coupler). The block diagram for the BIU is given in Figure 54.

The BIU is "smart", containing a processor with local memory, a "state information memory" (SIM), a FIFO buffer where messages are stored waiting to be transmitted, and transmitter/receiver units (one or more for each bus) with their own input and output buffers.

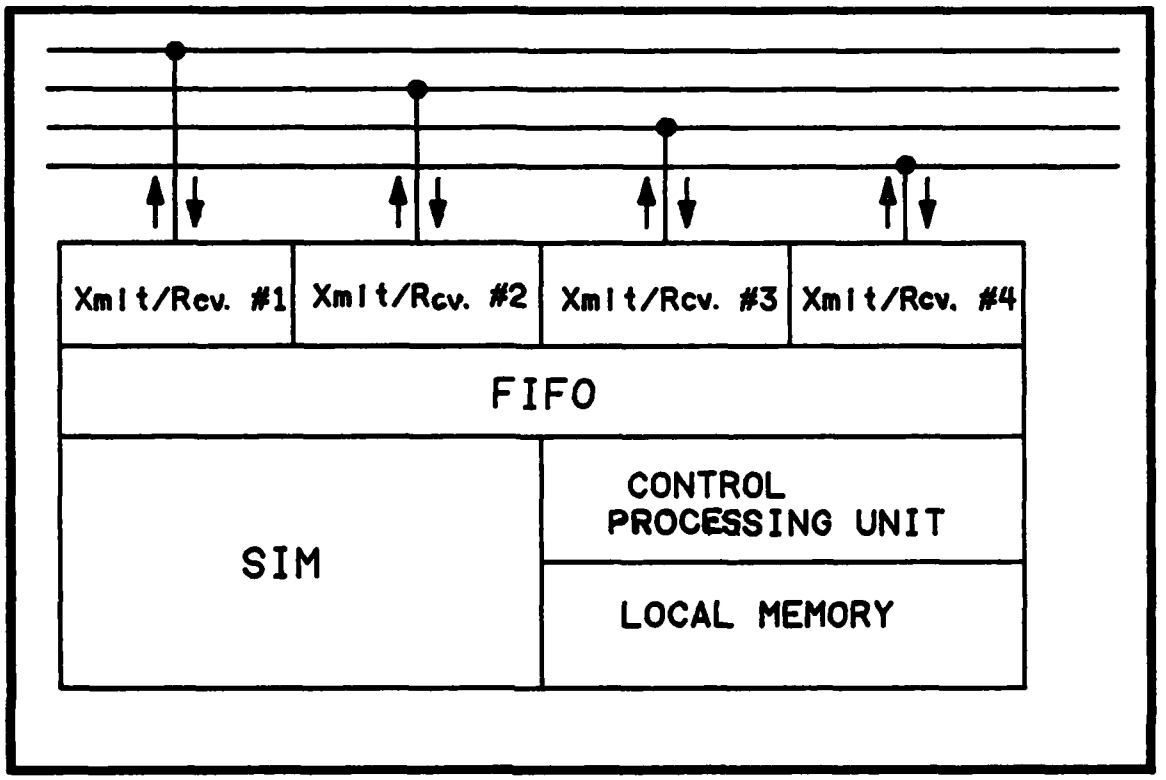


Figure 54. Bus Interface Unit Block Diagram

The fact that a BIU can transmit and receive simultaneously on different buses, along with the use of the transparent contention concept and a special arrangement for input and output buffers, allows the earlier mentioned features of acknowledgments in broadcast mode and the transient error self-correcting capability, as will be explained later.

Figure 55 illustrates one out of the four transmitter-receiver units.

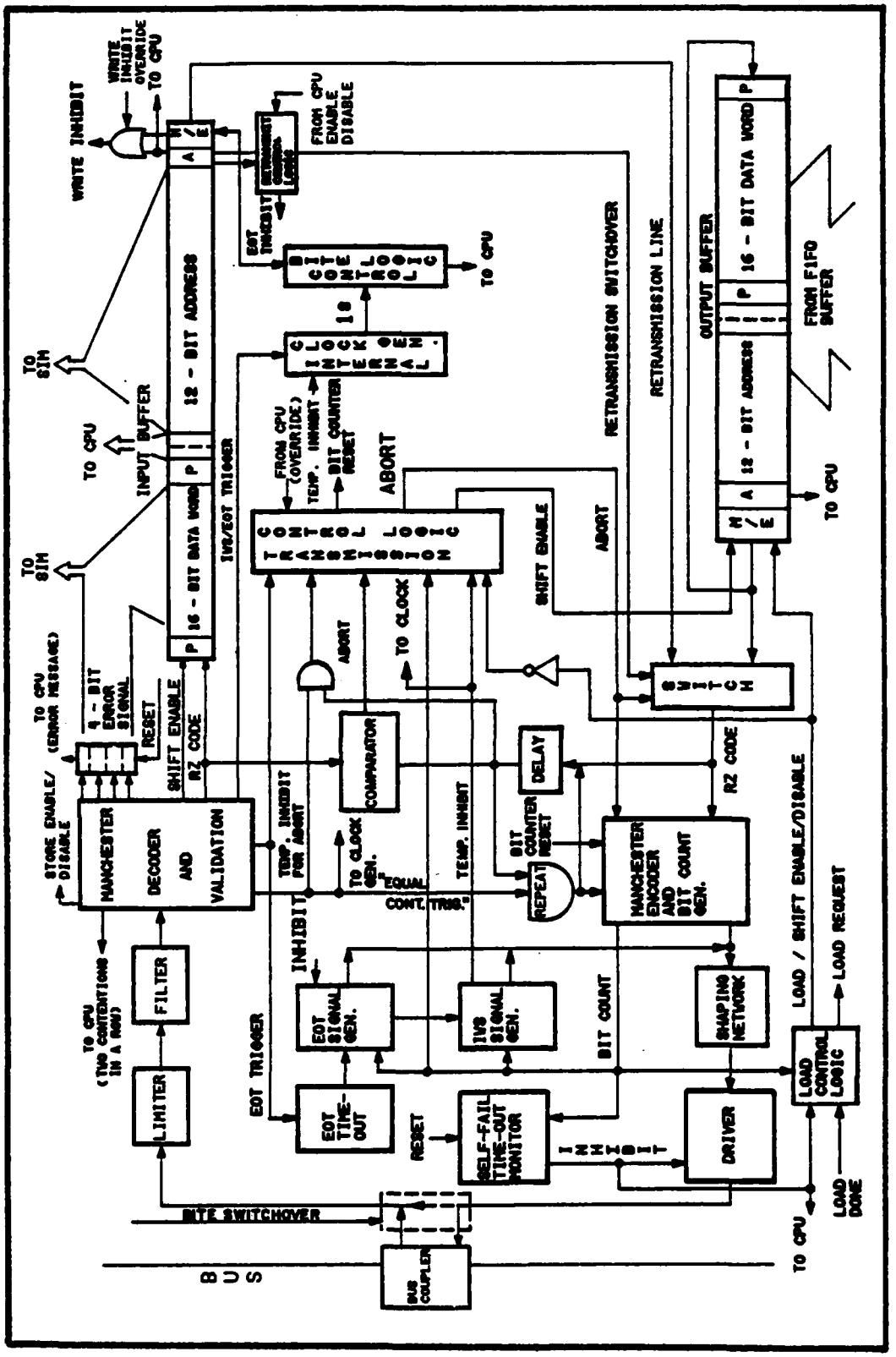


Figure 55. The SMS Transmitter-Receiver Unit

The receiving channel consists of a limiter, a filter, a Manchester decoder and validation circuit, and an input buffer in a special serial-in, serial/parallel-out arrangement.

The output channel contains a single-message parallel-in, serial-out rotating buffer, a retransmission switch, a Manchester encoder and bit count generator, a shaping network, and a bus coupler driver.

In normal process, the attached subsystem places messages into the output FIFO buffer which is common for all of the four transmitters. The BIU's processor distributes the messages from the FIFO buffer among the "empty" single-message transmitters' output buffers and the messages are ready to be transmitted. Each of the transmitters in a BIU has a different message in its buffer and all can transmit independently and simultaneously, but over the different buses as connected. Thus, the discussion that follows is the same for any of the four buses.

When an EOT occurs on the bus, the Manchester decoder will trigger the internal clock generator. The transmission control logic will at the same time eventually remove any existing abort signal and will apply the shift enable signal. This will allow the content of the output buffer to be clocked through the switch, encoder, shaping network, and driver out onto the bus. Each bit sent out is also delayed

before being applied to the comparator. This takes into account the time needed for the same bit to arrive from the other side of the comparator circuit. Any discrepancy between comparator inputs will cause an abort signal for the transmission control logic, and the encoder, as well as the main switch will be blocked.

While clocking out the message, the output buffer will at the same time rotate its content back into itself. When an abort occurs, the transmission control logic will continue to allow rotation of the output buffer content until the message is completely positioned for retransmission. Note that if there was no abort signal, this means that the message won the competition for a bus and the message has been completed. However, the same message is still in the output buffer and can be retransmitted if needed.

If "equal contention" occurs, the Manchester decoder and validation circuit in the receiving channel will generate a "one" for the Repeat AND gate and temporarily (for one bit time) remove clocks, which will inhibit receiving and transmitting. If the transmitter was trying to put a "one" on the bus when the "equal contention" occurred, the repeat logic will cause this "one" to be retransmitted without counting it as a new bit. At the same time, the "equal contention" trigger and delayed "one" will override the abort signal from the comparator. However, the encoder input

will be blocked for one bit time, allowing retransmission. If the transmitter was trying to put a "zero" on the bus when "equal contention" occurred, the abort signal will cause suppression of the transmission.

When the receiving buffer is full and if the message was neither an error signal nor data with acknowledgment request, the content of the buffer will be written into the SIM, but only if no error has been discovered in that transmission. At the same time, the four zero bits for error signal will reset the variable's book-keeping word.

The variable will not be written into the SIM if an error has been discovered; instead, the book-keeping word and the 4-bit error signal for the message on the particular bus are compared. If the error discovered in that transmission is of the same kind as already written in the book-keeping word, the CPU will flag the word and send it as the error signal message when the BIU has nothing else to transmit. This postponement of sending of error signal will give a busy terminal a chance to not transmit the error condition at all, because some of "free" terminals, if any, will do that in the meantime and the busy terminal will simply reset the word and remove the flag.

The book keeping word is always reset when a good message is received. In that way, the subsystem using a variable knows that the variable is properly updated and

accurate if the book-keeping word for that variable is all "zero". If one or more (different) errors are recorded, the subsystem is warned that the variable has not been updated, and will or will not use the variable as it is (depending on the subsystem and variable importance).

If the message was an error signal, writing into the SIM will be inhibited. Also, the control BITE logic will interrupt the CPU, and the CPU will decide either to perform BITE exercise if the variable was its own, or to reset the book-keeping word.

Acknowledgments in Broadcast Mode. The possibility to fully acknowledge the reception of a message transmitted in broadcast mode to unknown number of receivers is, probably, the greatest advantage of the Self-Managing Multiplex System. To the author's knowledge, there is no similar solution proposed so far. Usually, acknowledgments in a broadcast mode are considered in the literature to be either unfeasible or to be unreliable. However, in the SMS concept such acknowledgments are not only possible, but they are also used as a very efficient tool to fight transient errors. Thus, system integrity is enhanced. The technique is as follows:

Refer again to Figure 55. When a message with acknowledgment request is received, writing into the SIM is inhibitted, the CPU is interrupted, and the acknowledgment control

logic is enabled so that this time the input buffer clocks out its content through the main switch. Just before the retransmission from the input buffer starts, each of the acknowledgment control logic circuits (the retransmit control logic in Figure 55) will replace the acknowledgment bit in the buffer by a "zero", so that only one retransmission can occur at a time. Simultaneous retransmission by all terminals at the same time and on the same bus means that massive contention will occur. The mechanism of receiving, comparison, and eventually stopping of transmission is as usual and the net result is that receiver(s) which initially received bad message during the original transmission will have the good message when the retransmission is completed. Namely, in most cases transient errors will occur at the minority of receivers, so that the majority will win during the retransmission contention. The minority will stop their transmissions, while receiving the good message this time. This will correct any error(s) in the minority receivers' buffers.

Of course, there is a possibility that exactly half or even a majority of terminals will originally receive bad messages. In the acknowledgment retransmission process they will win, but note that the sender is also receiving the same retransmitted message. It knows what the good message is and in such case will simply repeat the message all over

again. This is possible because (1) the good message is still in its output buffer; (2) comparison on a bit-by-bit basis of the retransmitted and the good message has been already made by the sender during the retransmission process; and (3) the sender is only one which is responsible to send either an EOT if the good message has been retransmitted or an IWS if repetition is needed. This repetition can be done as many times as the system designer wishes - it depends on how the self-fail time-out monitor is set. The sender is, also, the only one which participates in the retransmission by sending the message from its output buffer. This obviously increases the number of good messages on the bus, but also allows the comparison of the good message with the retransmitted one in the usual manner through the sender's comparator. If an abort occurs for the sender's transmission logic, the good message will continue rotating back into the output buffer. Thus, when other terminals have finished the retransmission of an obviously bad message, the sender will have the good message ready and will send an IWS followed by the message.

Therefore, there is a very small possibility that transient errors affect the system performance, because the variables are written into the SIM only when it is almost absolutely sure that they are good in all of the receiving buffers. It is not likely that transient errors may occur

during the retransmission process, due to the fact that all of the terminals are transmitting at the same time and the signals on the bus are at or close to their maximal values.

One can argue that the technique has the same effect as if the sender in a classical system simply sends the same message twice or more times in a row. This is not true, because the sender without an acknowledgment never knows what really has been received at the other end of the bus regardless of how many times the message is repeated.

In addition, in most cases the SMS acknowledgment technique will at the first retransmission correct misinterpreted messages, if any. This is an obvious advantage over the standard correcting techniques that utilize CRC word(s) sent at the end of the message. Namely, the use of horizontal and vertical parity bits over the single words and the whole message, respectively, is senseless in general if detecting and correcting of more than three incorrect bits is required. In contrast, with the SMS concept as many bits as the whole message can be originally received wrong, but all the errors will be automatically corrected during the acknowledgement process.

There is no other technique which is capable to provide such a feature in more efficient way.

On the other hand, the inherent priority that the presence of light has over its absence, makes the SMS con-

cept, employed with optical links, more efficient in the normal operation, but less efficient in the self-correcting acknowledgment mode, when compared to the same system with a twisted-shielded pair bus. Namely, in the normal operation an "equal contention" will never occur on an optical bus. The bits that are "one" (presence of light) always win over the "zeros", regardless of how many terminals are trying to put this "zero" (absence of light) on the bus. This fact allows variables to be named in a prioritized order, which may be very important for some systems (trigger messages could be transmitted as soon as generated). However, since the majority does not rule on an optical bus, the self-correcting feature of the broadcast acknowledgment is useful only for the bits that were originally sent as "ones". In such a case, even if all terminals received this bit as a "zero", the retransmission of "one" only from the sender's output buffer will correct all the receiving buffers. But, if a "0" was the originally sent bit and only one terminal has interpreted it as a "one", this terminal will win during the retransmission and will cause the sender to repeat the entire message. Although it is not likely that the absence of light will be interpreted as its presence, it is still possible and the message may be required to be repeated. However, since the bus bandwidth is not a critical factor when optical links are employed, the more often repetition is a well

justified trade-off for enhanced reliability through the sender's assuredness that all receivers have received the accurate message.

System Network Topology. Figure 56 depicts a possible network topology of the hypothetical avionics system. The configuration is highly hierarchical in nature, decoupling the system functions. This possibility for functional partitioning is another feature of the SMS technique, allowing a system design without the possibility that errors in non-critical subsystems propagate and affect the flight-critical functions. Note that there is no limitation for growth of the system. Addition or deletion of subsystems, as well as software changing or updating, does not affect other subsystems. The same architectural concept can also be used down to the lowest level when there are no subsystems attached to the BIUs - the processors in the BIUs are performing the computational and other tasks.

A BIU is built by plugable, separated units. Such units can be used anywhere in a system, even at the lowest (sensor subsystem) level. This means a massive, cheap production. The main goal of any standardization would be achieved in that way, avoiding the trap of technological dependent standardization which can freeze possibilities for improvements as technology advances.

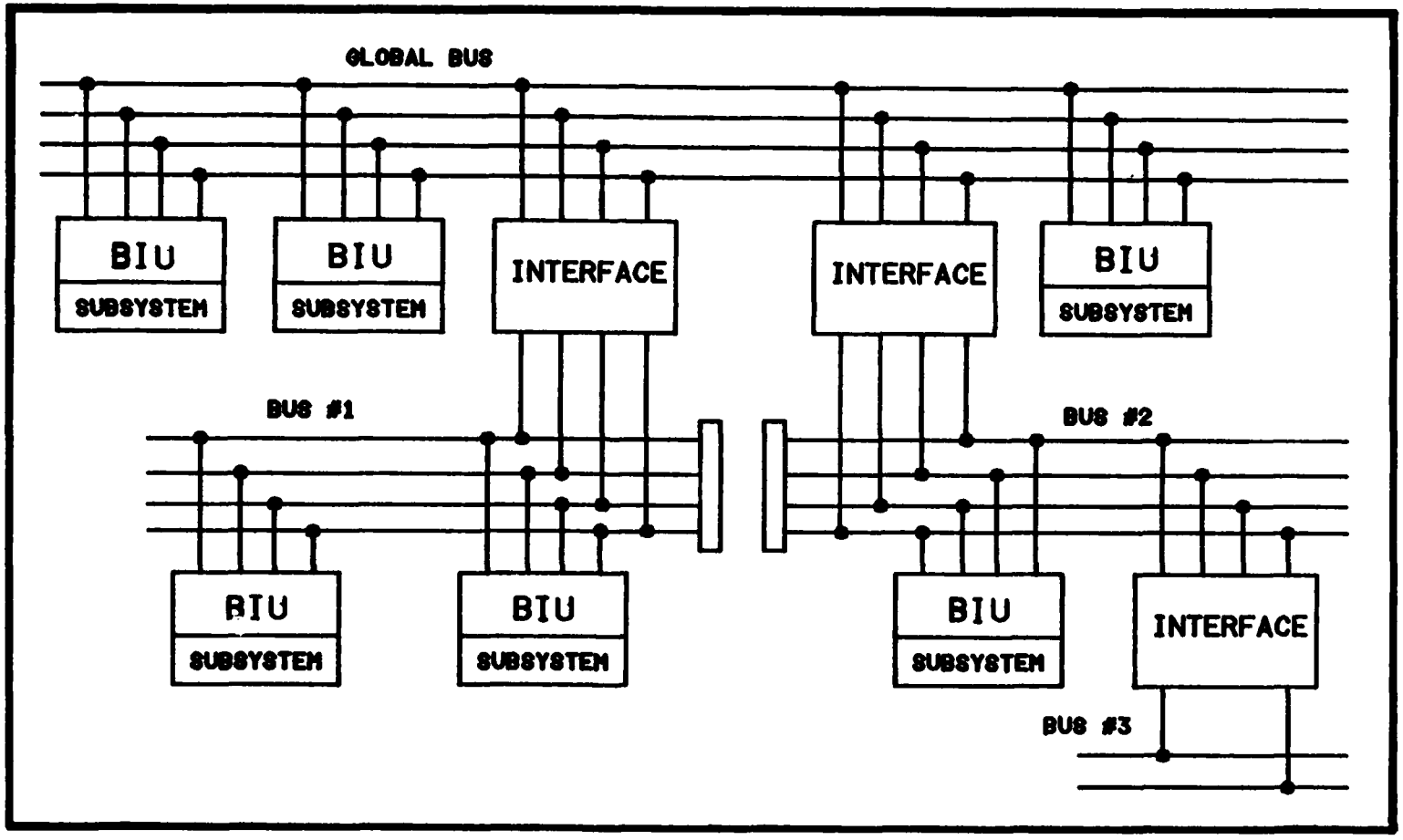


Figure 56. The SMS Network Topology

The interface for inter-bus communication is also built by the same basic units (Figure 57) and the processor in this case may be programmed to pass in either direction only the variables needed by subsystems attached to upper and/or lower levels. In such a case only FIFO buffers which are big enough to accommodate all incoming variables would be provided, without a need for the SIM.

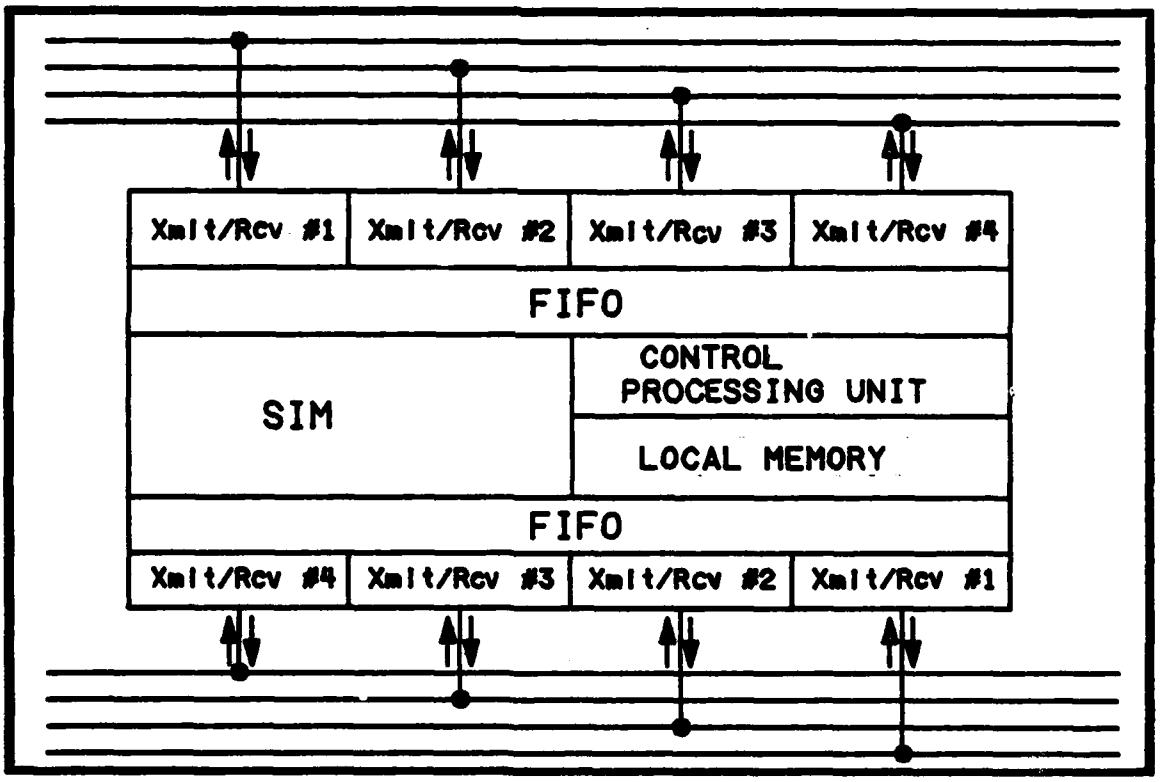


Figure 57. Inter-Bus Interface Unit

Redundancy in the System. The number of buses at any level is the designer's choice, as well as the number of redundant BIU's transmitter/receiver units attached to the

same bus(es). This redundancy for the BIUs is important when subsystems of greater complexity are involved. Namely, while the CRMmFCS technique of "task volunteering" (pp. 132-133) provides at the lowest, subsystem level that there is no way for faulty BIU to participate in message receiving, at the higher levels this technique is not suitable because complex subsystems cannot be replicated as many times as the task volunteering technique requires. This fact hides a possible dangerous consequence: a faulty transmitter/receiver unit in a BIU can cause that a message over a bus is missed. It is true that distribution of the messages over different buses is a pseudo-random process, such that the missed message will be refreshed very probably at next sampling time by the transmission over another bus where the BIU has a healthy transmitter/receiver unit. In that way, if a missed message has caused error propagation due to miscalculation based on the old value of the missed variable, in the next time frame it will be calculated correctly and the correction will propagate through the same path, recovering the system. However, having the assumption of cheap BIUs, it is better to have as many redundant transmitter/receiver units receiving messages from the same buses as is affordable. The units can then use the "volunteering" technique to compete for participation in transmitting/receiving within given time frames. The faulty units will be simply removed from the system by

giving up from the "volunteering". This also gives the possibility that the BIUs processor (or other units by voting technique) can shut down the faulty unit if it fails to do that by itself. (Continuous self-checking is already explained: a unit always receives its own message over the bus and can monitor itself).

Another technique at the designer's disposal is to reconfigure the system if all of the available receiving units for a bus are faulty. The BIU's processor could keep two additional tables in its local memory. One table contains all of the variables the terminal is reponsible to send out as generated by the attached subsystem, and the second one will contain a list of variables upon which the subsystem depends. In the case that all transmitter/receiver units for a bus are faulty, the terminal will simply send over other buses the list of variables it depends on. In that way, the other terminals will be informed about variables that should not be sent any more over the particular bus. The terminals generating these variables will append an inhibit flag for the bus to the variables in their own generating tables and will send these variables over other bus(es). This allows that the bus can still be used for other variables, without affecting the system performance.

The number of buses used in a system is determined based on the system requirements. A general guide should be

that the system will be safe if the number of faulty buses is less than or equal to one-half of the initially available buses. In the case of the hypothetical avionic system requirements as defined at page 149, this translates into the four buses required for the system. Namely, if 210 different variables are to be exchanged with an average update rate of 50 times per second, the data transfer loading is

$$210 \times 50 \times 40 = 420{,}000 \text{ bits/second}$$

where 40 bits are required to transfer a message containing a single data word (page 157). Assuming the worst case, when two-thirds of the variables are required to be acknowledged, the load becomes 700,000 bits/second, which increases to 840,000 bits/second if the error rate is so high that every other acknowledgment fails and the messages must be repeated with further acknowledgments.

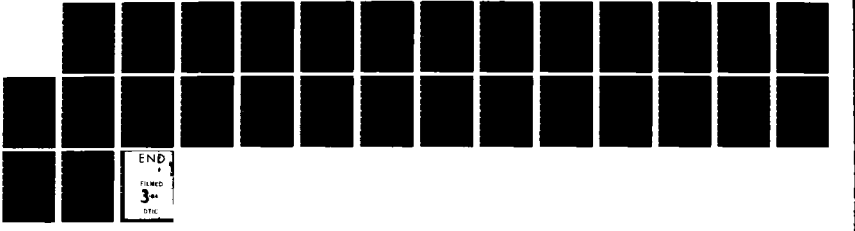
If the rest of one-third of the variables are to be covered by the book-keeping error signals that are transmitted for each variable five times per second, then the load increases by additional $70 \times 40 \times 5 = 14{,}000$ bits/second and the total load becomes 854,000 bits/second. The difference to 1 Mbps of 146,000 bits/second is probably enough for the purposes of trigger messages transfer, system reconfiguration, the losses due to "equal contention", etc.

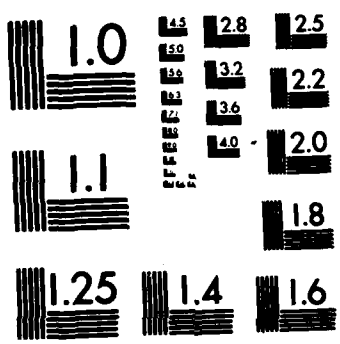
Assuming subsequent system growth up to twice the load initially anticipated, the final result will be 2 Mbps of

AD-A138 151 A STUDY OF OPTIMAL COMPUTER NETWORK ARCHITECTURE FOR DIGITAL AVIONICS SYSTEMS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. M F KRILIC

3/3

UNCLASSIFIED DEC 83 AFIT/GE/EE/83D-36 F/G 17/2 NL





gross load in the system. That means that two always operational buses are needed in the system, which results in the total of four buses, as has already been said.

Of course, the above calculation is very rough and based on assumptions that may not be true for an actual system. Note, however, that the system is insensitive to loss of one bus, completely safe if two buses fail, and still usable if even three out of four available buses are not operational (provided that message acknowledgments are reduced).

Bus Length. The signal propagation delays limit the maximal allowable bus length. The same is true for any concept, but in the SMS technique further complications are involved due to the signal summation that results in quite complex waveforms on the bus. There is a large number of different waveforms that can occur as a consequence of different signal propagation delays and attenuations, allowable zero crossing deviations, and different rise and fall times for the generated signals. The situation is further aggravated by the changing number of competing terminals.

A complete investigation of this phenomena requires further research encompassing the total spectrum of possible waveforms. An additional thesis effort, implementing the Self-Managing Multiplex System via computer simulation techniques is highly recommended.

However, a limited number of cases that are investigated within this work led to the conclusion that there are no problems at all when the concept is implemented with shorter buses, such as the F-16 multiplex bus. If a longer twisted-shielded bus is needed, then the existing technology for the Manchester coding, decoding, and validation must be improved. As a first, rough estimate for the generated signals, the allowable zero crossing deviation should be limited to ±10ns as well as the rise and fall times to the maximum of 100.0 ns. Also, the receivers must be capable to operate with the incoming signals with a maximum zero crossing deviation of ±200.0 ns. With such characteristics, the allowable bus length would be up to 150 feet, which is more than 8.5 times longer than the F-16's bus, for example.

The maximum of 150 feet for a twisted-shielded bus is a conservative estimation, because none of the extenuating circumstances is considered. Namely, the filtering effect of just a few feet of cable significantly reduces the problems caused by the unfiltered waveforms (Ref 41:108). The twisted pair shielded cable is essentially a distributed low-pass filter. In addition, the limiter and the filter within each receiver (Figure 55, page 165) are intended to shape the incoming signals, so that the information can be extracted from the direction of voltage transition. This transition is only what is important for the Manchester decoding. Also,

even current hardware is capable of handling signals varying in a wide range from square to sine waves (Ref 65:27).

Note that the waveform deviation problems almost do not exist in the acknowledgment mode. The simultaneous transmission of all terminals makes the signal distortion occur at higher voltage levels, which will be cut off by the receivers' limiters. The greater the majority (which is normal in the acknowledgment mode), the smaller the resulting signal distortion. This fact adds more to the acknowledgment mode importance.

Optical buses do not suffer from such problems, at least not to the same extent. The bus length is limited only by the requirement that the light presence within one bit time frame must be continuous and must not extend into the next bit time frame (to overlap with a new bit). Using the convenient rule of thumb that the velocity of light is one foot per nanosecond, it is easy to see that above requirements translate into a maximum bus length of up to 500 feet.

Number of Terminals. The electrical characteristics of terminals that are feasible with today's technology also expose some limitations on the number of terminals connected to a twisted-shielded pair. In order to calculate the allowable number of terminals per bus for the hypothetical system considered, a couple assumptions must be made first:

(1) The receivers' dynamic range should allow all terminals to respond to Manchester-like signals with peak-to-peak amplitude in the range of 0.86 volts to 14.0 volts for transformer coupled stubs. The range is chosen as for ordinary MIL-STD-1553 terminals (Ref 65:27), so that as much as possible existing technology can be implemented.

(2) The line power loss is up to 1.5 dB/100 ft, as defined in the MIL-STD-1553 (Ref 65:22). If the maximum of 150 feet in the bus length is assumed, then the worst case total power loss on the bus is 2.25 dB.

(3) Referring again to the published data for existing systems (Ref 41:47 and 5:239), let the transmitter output impedance be equal to the characteristic impedance (70 ohms) of the bus, and the receiver input impedance be 5Kohm resistive component at 1MHz.

Capacitor coupling is used at the transmitter and the transformer coupling is used at each line receiver.

As has already been discussed (pp.20-25), the transformer coupling will provide very good common-mode,noise rejection while maintaining the bus in a balanced condition. The capacitive coupling at the transmitters is chosen for this system in order to provide a low impedance coupling to the bus, while having the necessary dc isolation. The open collector transistors in the bus-coupler drivers (Figure 55, p. 165) are switched off when not being in the transmitting

mode, isolating the transmitters from the bus. In fact, such capacitor coupling has been utilized in a current system for which enough data was available and this is the reason that it is proposed for this system, too. For an actual implementation, the transformer coupling for both transmitter and receiver is still more preferred.

With the described configuration of transmitter and receiver coupling, the experimental results published by the TRW Systems (Ref 5:239, Fig. 3) showed that the Manchester signals voltage level received at the other end of a 500 feet long twisted-shielded bus, with a load of 64 terminals and total power loss of 7.5 dB, was reduced only by one-half with respect to the level of transmitted signals. Having a considerably shorter bus (150 feet) and the receivers' dynamic range assumed for the system described in this chapter, it seems reasonable to assume that all transmitters are to provide 1.5 volts, peak-to-peak amplitude for output signals (for a comparison, this is the half of the B-52's voltage levels - Ref 44:88, Table 9). Such low amplitude of the output signals will decrease the power dissipation at the transmitter's bus coupler driver, while still being above the lowest allowable signal level at the farthest receiver site (150 feet), even in the worst case when only one terminal is transmitting.

Thus, if 64 terminals are assumed, the maximal voltage

on the bus will be ±48 volts in the worst case when all transmitters are transmitting the same bit at the same time. This is less than 7 times as high as the upper limit of the receiver's dynamic range. Today's technology allows the receivers' active limiters to achieve such voltage attenuation with relative ease (i.e., without extensive cooling requirements).

Block and Trigger Message Transfer. The illustrative design of the transmitter/receiver unit (Figure 55, p. 165) only allows messages with a variable address and one data word to be transmitted. Block-messages transfer transmission, if needed in an actual implementation, would require a more complex design.

The optical links implementation of the SMS concept allows a very efficient trigger message transfer. Thanks to the inherent priority that the presence of light ("one") has over its absence ("zero"), the proper addressing of trigger messages will allow such messages to win any competition. That means that a trigger message will be transmitted as soon as it is generated. It is possible, even, to order trigger messages themselves, although it is not likely that two or more trigger messages will ever occur on the bus at the same time.

With the twisted-shielded pair, ordering is not possible, because the majority rules on such a bus. If the system

software complexity (to be discussed later) and the allowable trigger message latency do not allow the worst case when a trigger message waits until it is alone on the bus, then hardwired connections for the trigger messages is recommended. If an actual system would require extensive hardwiring for such connections, then the solution is to dedicate separate redundant multiplex buses for all trigger messages in the system. The messages will compete for the bus in the usual manner, but again, it is not likely that two or more of such rare messages may occur at the same time.

Bus Efficiency. Calculated as defined for this report, the bus efficiency for the SMS concept is constant in a normal operation and equals 40 percent. This is twice as much as the MIL-STD-1553 efficiency, when compared with messages containing one data word. The SMS system is also capable of handling block-message transfers, although this is not employed in the conceptual design shown in this chapter. If this was the case, the system would reach bus efficiency of 75% with just 16 data words in a message, while a MIL-STD-1553 bus has such efficiency only as its maximum (32 data words within single message).

While a SMS optical bus retains the 40-percent constant efficiency in normal operation, the possibility of "equal contention" causes that a twisted-shielded bus may have this efficiency in the range from 34.8 to 40 percent. The lower

limit corresponds to the worst case of six such contentions for 64 terminals trying to transmit at the same bus allocation signal.

Assuming, again, that two-thirds of the variables are acknowledged with each other acknowledgment repeated and the book-keeping words for the rest of variables, then the SMS bus efficiency drops down to 9.7 percent due to significant overhead of doubled acknowledgments. Note, however, that the assumption represents an extremely high error rate that is not likely to occur in an actual system. The assumption is made just because of eventual side effects with a twisted-shielded bus, which should be thoroughly investigated within further research, as has been recommended.

Bus Utilization and Message Latency. The bus utilization and message latency for the SMS concept cannot be calculated exactly, because the actual system specification requirements are necessary for such calculations. Instead, a discussion about another feature of the SMS concept follows:

The subsystem connected to its BIU provides data to be transmitted only when the data has been changed since the last time it was transmitted. In that way, the system becomes an information-only transfer system, not just a data-transfer system. Namely, in conventional systems which are data-transfer systems, a bus can be just as busy sending or receiving old data already known by all terminals as trans-

ferring important information (Ref 61:690). Information-only transfer will increase considerable the data throughput.

Repetitive updates of static information consume much of the available bus bandwidth. A more meaningful characteristic to calculate for an information-only transfer system would be the bus utilization defined as ratio of useful data transferred versus the total number of bits used (Ref 90:43). A comparison, based on this bus utilization ratio definition, between all of the systems described in this report and the Self-Managing Multiplex System, would very probably significantly reduce the performance figures of all other systems since much of data they transfer is redundant.

Of course, the SMS bus time must still be allocated for the worst case when all variables do change, but the bus will become idle only when all of important information already has been sent. This is far better than the case when an important message waits for useless data to be sent first.

Obviously, a significant decrease in information latency (not just message latency) is the primary advantage of the information-only systems. Thus, some in-line delay inherent to SMS hierarchical topology due to the inter-bus communication (Figure 56, page 175) is more than sufficiently compensated by the benefits of information-only transferring.

Software Interface. The software engineering aspects are on the other side of coin and are as important as the architectural and hardware support considerations.

One could argue that there is the possibility for a SMS variable to be ready in the output buffer but wait forever, constantly losing the competition. However, the worst case calculation made on page 179 for the hypothetical system, shows that each variable has its chance 50 times per second on the average, which is exactly what was needed for the system. In normal operation, this total load for the worst case is less than 50 percent of the available 4 Mbps in a SMS system with four twisted/shielded buses. Obviously, each variable will sooner or later get its chance due to the fact that the buses are never idle if there is some data to be sent. The situation is further improved by the information-only transfer capability.

The probelm is how to assure that the variables occur on the bus when needed by the other subsystems (the SMS approach assumes only broadcast mode, and a source never asks for information).

In a command/response system, as well as in polling techniques, the capability for information request is paid for by system inflexibility and by high message overhead.

A very useful approach to the problem is known as the "quantized software" technique (the "software millimodules"

in the CRMmFCS, p. 130-131). It gives an opportunity to know almost exactly when a variable will be ready for transmission, but such an advantage is paid for by wasting some amount of time when a processor idles waiting for the next time milliframe.

The "event-driven" systems require processor interruption when a new variable arrives and uses this "event" as a trigger for the task execution. Although very suitable for real-time systems, this technique requires more complicated hardware.

For the SMS concept it is assumed that the subsystems are responsible to send only the data that has changed since the last time sent. This means that the other subsystems will use the value already stored in the SIMs. In fact, each subsystem is constantly executing its program as a recursive loop by first performing all necessary readings out of the SIM, then executing the program as a whole or partially as a module. The results are then compared with the previous values and will be put into BIU's output FIFO buffer only if the variable has been changed since the last time updated. The program execution then proceeds to the next module. Of course, it may happen that a variable needed for a previous software module arrives with a new value while the processor is already executing subsequent module(s), but that is not of critical importance unless the time expired from when the

variable has been generated to the time when it is used is important (as in navigation systems, for example). This problem, however, is already solved by time-tagging to such data (Ref 49:220) and the same solution is also assumed to be used in the SMS concept-based systems. What is important for the system designer is to assure that the entire system software is designed in a way giving no possibility that any variable can be transmitted more than once before actually used by each of the corresponding subsystems. If such a possibility is unavoidable, the solution is in "quantized software", especially at lower, subsystem levels where sacrificing the bus throughput (which usually is not critical at these levels) for simpler software is well justified.

Anyhow, both possibilities do exist and are, again, open to the designer's choice.

VI. Conclusions and Recommendations

This report has been intended to define the "optimal" computer network architecture for future digital avionic systems. The author identified more than five hundred references that show how much attention the subject has received in recent years. The survey of available literature as given in Chapter II, is one of the results of this effort and can help further research in the field.

In order to define the characteristics of optimal architecture, the current theoretical knowledge is documented in Chapter III, where an attempt is made to sort out the essential figures of merit of any avionics system architecture. The key issues, as defined in that chapter, are used in Chapter IV to evaluate fourteen different approaches to the same problem of data multiplexing in avionic systems. In that way, the current system design base is also documented within a single report.

Chapters III and IV represent the second result of the study and can be eventually used in further research to establish some ordered and weighted criteria for evaluation of any architecture that may be proposed in the years to come. The only similar attempt, made in the reference 49, is not elaborated to the extent which allows an automated evaluation (by a computer simulation technique, for example).

Such research is recommended and this author hopes that the Chapters III and IV of this thesis can serve as a starting point should the recommendation be accepted, in spite of the conclusion (page 15) that the uniquely defined criteria to evaluate all possible solutions can hardly find the universal application.

The characteristics of the optimal computer network architecture for future digital avionic systems are defined in Chapter V.

Briefly, the optimal architecture must reflect the results of the successful current system design base, involving advantages of proven techniques. The architectural approach should be reliable, simple, and general enough to be standardized for use throughout the aerospace industry. The standardization should be possible from the lowest avionics subsystem up to the global system level. The architectural philosophy should also leave the system designer free and technologically independent to implement the concept according to the particular subsystem or overall avionic system requirements, allowing at the same time a possibility for subsequent growth and multi-mission applications of the system with relative ease.

The author believes that the described Self-Managing Multiplexing concept approaches an "optimal" computer network architecture for avionics systems, as defined above.

The concept includes a variety of solutions already implemented, but integrated in somewhat new ways, giving features that cannot be found in the other techniques. It also offers great flexibility to system designers, without technology dependence, and enhances the fault-tolerance to a high level.

The normal procedure of dynamic reconfiguration requires no emergency reconfiguration in the case of failures. The technique is well-matched to distributed architectures and the hierarchical topology of the system allows functional partitioning both in hardware and software.

Fast and dynamic, the transparent contention bus reallocation scheme allows the bus utilization in any sequence and with any percentage of bandwidth. The information-only transfer concept adds more to these features.

The benefits are not achieved without cost: high level hardware support is needed. However, it could be considered as an acceptable trade-off, especially if the concept finds its place in future avionics standards and allows massive, cheap production. This will also simplify logistics by common modules maintenance. In addition, as pointed out several times throughout this report, more sophisticated hardware combined with a high bus bandwidth in an advanced architecture will raise the system performances and decrease software complexity and costs at the same time.

The Self-Managing Multiplex System is introduced only conceptually and requires further engineering design, refinements and implementation trade-offs on a hot-bench. A laboratory testing of variety of different ideas integrated in this concept is desirable, but a computer simulation of a SMS-based system with twisted-shielded bus should be performed first to lower the risk of hardware investment in a new technique. A complete investigation of the total spectrum of possible bus waveforms is highly recommended and can be performed within additional AFIT thesis work.

The intriguing features of the newly introduced acknowledgment technique in a broadcast mode with automatic self-correcting of burst transient errors, could be very useful if proven in practice. As far as the author's knowledge goes, there is no similar proposal made so far and there is no other technique that offers so efficient a way to fight transient errors; therefore, feasibility testing is really necessary. The technique can be tested even outside of a SMS-based system, on the CRMmFCS hot bench that exists in the AFWAL/WPAFB. Modifications to the bench are straightforward and, at least, this result of the study seems to deserve some sort of testing in practice as soon as possible.

Bibliography

1. Adams, E.S. and B. Clausing. "Distributed Avionics Processing Using ADA," Proceedings of the National Aerospace Electronics Conference, 979-983. Dayton, Ohio: Dayton Section IEEE. May 1983.

2. Allen, R. et al. System Avionic Architectures for RPVs. Technical Report, Texas Instruments, Inc., Dallas, Texas. April 1977. (AD-A041502).

3. Air Force Systems Command - ENASD. The MIL-STD-1553 Multiplex Applications Handbook, WPAFB, Dayton, Ohio, May 1980. (AD-A119529).

4. Barnett, P.R. "Implementation of Broadcast Messages and Acyclic Data Transfer Techniques in a Multibus Based Avionic System," Proceedings of the National Aerospace Electronic Conference, 625-632. Dayton, Ohio: Dayton Section IEEE. May 1983.

5. Bello, L.M. and T.C. Berg. "LSI Data Bus for Avionics," Proceedings of the National Aerospace Electronics Conference, 238-244. Dayton, Ohio: Dayton Section IEEE. May 1973.

6. Boose, E.F. "Lessons Learned Through a MIL-STD-1553 Time Division Multiplex Bus," Proceedings of the National Aerospace Electronics Conference, 634-641. Dayton, Ohio: Dayton Section IEEE. May 1975.

7. -----. "Shielded Twisted Pair as a Transmission Medium in a Multisubscriber Time Division Multiplex System," Proceedings of the National Aerospace Electronics Conference, 648-655. Dayton, Ohio: Dayton Section IEEE. May 1975.

8. Booton, W.C. A Conceptual Definition Study for a Digital Avionics Information System (Approach I), Volume I. Technical Report, General Dynamics Convair Aerospace Division, Fort Worth, Texas. October 1973. (AD-771736).

9. -----. A Conceptual Definition Study for a Digital Avionics Information System (Appr. I),Vol. II. Technical Report, General Dynamics Convair Aerospace Division, Fort Worth, Texas. October 1973. (AD-771756).

10. Brodnax, C.I. A Conceptual Definition Study for a Digital Avionics Information System (Approach II), Volume I. Technical Report, Texas Instruments, Inc., Dallas, Texas. March 1974. (AD-780581).

11. -----. A Conceptual Definition Study for a Digital Avionics Information System (Approach II), Volume II. Technical Report, Texas Instruments, Inc., Dallas, Texas. March 1974. (AD-780582).

12. -----. A Conceptual Definition Study for a Digital Avionics Information System (Approach II), Volume III. Technical Report, Texas Instruments, Inc., Dallas, Texas. March 1974. (AD-780583).

13. Bussell, J.F. "A Concept of Digital Avionics," Proceedings of the AGARD Conference "Inertial Navigation Systems and Components", 283-295. Braunschweig, W. Germany, May 1968.

14. Calhoun, M.D. and B. Khatirzad. A Study of Avionics Time Division Multiplex Bus Simulation. Technical Report, Mississippi State University, Mississippi State, Mississippi. December 1980. (AD-A097375).

15. Chien, R.T. and L.J. Peterson. Optimized Computer Systems for Avionics Applications. Technical Report, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois. February 1980. (AD-A089570).

16. Cook, M.J. et al. Digital Avionics Information System (DAIS): Development and Demonstration. Technical Report, TRW Defense and Space Systems Group, Redondo Beach, California. September 1981. (AD-A107906).

17. Cotter, G.E. et al. "Avionics Multiplexing with Smart Terminals," Proceedings of the National Aerospace Electronics Conference, 672-678. Dayton, Ohio: Dayton Section IEEE. May 1975.

18. Courter, J.G. "Application of Multiplexing to the B-1 Aircraft," Proceedings of the National Aerospace Electronics Conference, 351-354. Dayton, Ohio: Dayton Section IEEE. May 1973.

19. Crossgrove, W.A. and J.D. Blair. "Categorization of Data Bus Architectures for Avionics and Flight Control Systems," Proceedings of the National Aerospace Electronics Conference, 406-411. Dayton, Ohio: Dayton Section IEEE. May 1980.

20. -----. "Distributed Systems: The Next Integration Method," AIAA Paper No. 77-1484. (1977).

21. -----. "Development and Applications of MIL-STD-1553," SAE Technical Paper Series, No 801142. October 1980.

22. Cunningham, John M. Investigation of Functional Commonality of Avionics Systems in Naval Aircraft. Technical Report, Naval Air Development Center, Warminster, Pennsylvania. September 1981. (AD-A105503).

23. Czuchry, A.J. et al. MID-1980's Digital Avionics Information System Conceptual Design Configuration. Technical Report, Dynamics Research Corporation, Wilmington, Massachusetts. July 1976. (AD-A032137).

24. Davies, D.W. et al. Computer Networks and Their Protocols. John Wiley & Sons, 1981.

25. Davis, Richard T. "Solutions Considered for Future High Speed Military Bus," Defense Electronics 5 (15): 48-62. May 1983.

26. Denery, D.G. et al. "Design, Development, and Flight Test of a Demonstration Advanced Avionics System," NASA Technical Memorandum 84323. January 1983.

27. Denny, M. and T. Planbeck. "Active Memory Subsystems: An Architectural Approach to Design Recoverability," Proceedings of the National Aerospace Electronics Conference, 1279-1282. Dayton, Ohio: Dayton Section IEEE. May 1980.

28. Dunn, G.L. and P. Leong. "Conceptual Design of an Integrated Power and Avionics Information System," Proceedings of the National Aerospace Electronics Conference, 954-962. Dayton, Ohio: Dayton Section IEEE. May 1981.

29. -----. "Real Time Monitoring and Analysis of a MIL-STD-1553A/B Multiplex Data Bus," Proceedings of the Third AIAA/IEEE Digital Avionic Systems Conference, 145-149. Dayton, Ohio. October 1979.

30. Dworkin, B. "The Avionics Tree - A Systematization of System Avionics," Proceedings of the National Aerospace Electronics Conference, 89-95. Dayton, Ohio: Dayton Section IEEE. May 1976.

31. Elmore, Vicki L. et al. "Squeezing 1553 Bus Into the MSRAM," Defense Electronics, 104-113. August 1983.

32. Engel, H.E. et al. Digital Avionics Information System (DAIS): Current Maintenance Task Analysis. Technical Report, Dynamics Research Corporation, Wilmington, Massachusetts. October 1976. (AD-A035683).

33. Friedman, S.N. "MIL-STD-1553 Dynamic Bus Controller/ Remote Terminal Hybrid Set," Proceedings of the National Aerospace Electronic Conference, 639-644. Dayton, Ohio: Dayton Section IEEE. May 1983.

34. Gaarder, T.N. The Theory of Digital Processing of Stohastic Signals for Use in Avionic Systems on-Board Air Force Aircraft. Scientific Report, University of Hawaii, Honolulu, Hawaii. March 1971. (AD-719822).

35. Gangl, Ervin C. "Avionic Systems Integration Using Digital Computers," Proceedings of the AGARD Conference: "Automation in Manned Aerospace Systems." Dayton, Ohio. October 1972.

36. -----. "Digital Systems Integration Techniques, the Key to Affordable Avionics," AIAA Paper No. 77-1492. (1977).

37. Gifford, C.A. "A Military Standard for Multiplex Data Bus," Proceedings of the National Aerospace Electronics Conference, 85-88. Dayton, Ohio: Dayton Section IEEE. May 1974.

38. Gregory, J.G. "System Architecture for Aircraft Avionic and Electrical Systems," Proceedings of the National Aerospace Electronics Conference, 30-35. Dayton, Ohio: Dayton Section IEEE. May 1973.

39. Griffin, Edward L. "Standards and Integrated Avionic Digital System Architecture," Proceedings of the Second AFCS Standardization Conference, 563-581. Air Force Systems Command, WPAFB, Dayton, Ohio. December 1982.

40. Gross, J.P. "Data Bus Techniques for Digital Avionics," Proceedings of the National Aerospace Electronics Conference, 230-237. Dayton, Ohio: Dayton Section IEEE. May 1973.

51. Kuo, F.F. Protocols and Techniques for Data Communication Networks. New Jersey: Prentice-Hall, Inc., 1981.

52. Lambert, J.W. and J.P. Green. A Comparative Study of Data Bus Systems. Technical Report, Intermetrics Inc., Cambridge, Massachusetts. Marc 1978. (IR-270 under contract N62269-77-M-5298 for NADC).

53. Larimer, Stanley J. "A Solution to Bus Contention in a System of Autonomous Microprocessors," Proceedings of the National Aerospace Electronics Conference, 309-317. Dayton, Ohio: Dayton Section, IEEE. May 1982.

54. -----. AFWAL/FIGC, WPAFB, Dayton, Ohio. Private conversation. 18 July 1983 and 30 August 1983.

55. -----. A Continuously Reconfiguring Multi-Microprocessor Flight Control System. Technical Report, Flight Dynamics Laboratory (AFWAL/FIGC), WPAFB, Dayton, Ohio. May 1981. (AD-A101412).

56. Leib, John E. "Application of 1553B to MSRAM - A System Look," Proceedings of the Second AFSC Standardization Conference, 295-304. Air Force Systems Command, Dayton, Ohio. December 1982.

57. Mahler, B.B. An Analytical Examination of Impedance and Signal Power Levels along a Multiplex Data Bus. Technical Report, Mitre Corporation, Bedford, Massachusetts. March 1975. (AD-A008498).

58. -----. A Comparison of Time Division Multiplex Data Buses Based on Computer Simulation. Technical Report, Mitre Corporation, Bedford, Massachusetts. May 1975. (AD-A012305).

59. Mason, R.C. et al. Digital Avionics Information System (DAIS): Development and Demonstration. Technical Report, TRW Defense and Space Systems Group, Redondo Beach, California. March 1979. (AD-A068438).

60. McCartney, R.I. and R.E. Phillips. "A Modular Approach to MIL-STD-1553 Simulation Support," Proceedings of the National Aerospace Electronics Conference, 872-877. Dayton, Ohio: Dayton Section IEEE. May 1981.

61. McDonald, J.E. "An Architecture for Event-Driven Real-Time Distributed Computer Systems," Proceedings of the National Aerospace Electronics Conference, 688-694. Dayton, Ohio: Dayton Section IEEE. May 1983.

41. -----. Techniques for Interfacing Multiplex Systems. Technical Report, SCI Systems, Inc., Huntsville, Alabama. February 1981. (AD-A101457).

42. Haley, A.L. "MIL-STD-1553 Validation Test Results," Proceedings of the National Aerospace Electronics Conference, 747-749. Dayton, Ohio: Dayton Section IEEE. May 1980.

43. Harris, R.L. and E.E. Jones. "Fault Tolerance Applications to Future Military System Avionics," Proceedings of the National Aerospace Electronics Conference, 412-419. Dayton, Ohio: Dayton Section IEEE. May 1980.

44. Hawkins, R.D. "Development Considerations for Real-Time Aircraft/Avionic Distributed Computer Control Systems," Proceedings of the National Aerospace Electronics Conference, 586-589. Dayton, Ohio: Dayton Section IEEE. May 1983.

45. -----. "Buffered Receptor, Avionics Integration Network (BRAIN) -- A Concept Proposed for Memory Managed Avionics," Proceedings of the National Aerospace Electronics Conference, 659-668. Dayton, Ohio: Dayton Section IEEE. May 1983.

46. Hecht, Herbert and Myron. Fault Tolerance, Reliability and Testability for Distributed Systems. First Interim Technical Report, SoHar Inc., Los Angeles, California. February 1983. (RADC-TR-83-36).

47. Husbands, C.R. "Microprocessors in Ariborne Time Division Multiplex Terminals," Proceedings of the National Aerospace Electronics Conference, 642-647. Dayton, Ohio: Dayton Section IEEE. May 1975.

48. ILC Data Device Corporation. MIL-STD-1553 Designer's Guide. Bohemia, New York, 1982.

49. Jensen, D.E. et al. The Impact of Wideband Multiplex Concepts on Microprocessor-Based Avionic System Architectures. Technical Report, Honeywell Ssytems and Research Center, Minneapolis, Minnesota. February 1978. (AD-A057878).

50. Klos, Larry C. "Protocol for Modular Avionics Architecture Serial Multiplex," Reported at the National Aerospace Electronics Conference. Dayton, Ohio: Dayton Section IEEE. May 1983.

62. McLaren, B.D. "B-1 Avionics Data Multiplexing System (AMUX)," Proceedings of the National Aerospace Electronics Conference, 205-212. Dayton, Ohio: Dayton Section IEEE. May 1975.

63. McSharry, M.E. and L.A. Smith. "Contention Information Trasnfer Systems Analysis," Proceedings of the National Aerospace Electronics Conference, 402-426. Dayton, Ohio: Dayton Section IEEE. May 1980.

64. Meyer, D.H. and D.R. Parr. "A MIL-STD-1553 Flexible Interface Device and Applications," Proceedings of the National Aerospace Electronics Conference, 618-624. Dayton, Ohio: Dayton Section IEEE. May 1983.

65. MIL-STD-1553B: Aircraft Internal Time Division Command Response Multiplex Data Bus. 21 September 1978.

66. MIL-STD-1760: Aircraft/Store Electrical Interconnection System. 1 July 1981.

67. Moran, John F. "A Bus Oriented Modular Avionic Architecture," Proceedings of the National Aerospace Electronics Conference, 698-702. Dayton, Ohio: Dayton Section IEEE. May 1980.

68. Morgan, R.D. and R. Bellem. "Pave Pillar: A Maturation Process for an Advanced Avionics Architecture," Proceedings of the Second AFCS Standardization Conference, 675-690. Air Force Systems Command, Dayton, Ohio. December 1982.

69. Mrazek, J.G. "Flight Test and Demonstration of Digital Multiplexing in a Fly-By-Wire Flight Control System," Proceedings of the National Aerospace Electronics Conference, 192-197. Dayton, Ohio: Dayton Section IEEE. May 1973.

70. Mueeoq, R. and J.L. Perry. "Multiplexing in Aircraft Subsystems," Proceedings of the National Aerospace Electronics Conference, 343-450. Dayton, Ohio: Dayton Section IEEE. May 1973.

71. Newcomb, R.W. Theory of Functional Networks and Microsystems for Avionics System. Scientific Report, Stanford University - Electronics Laboratory, Stanford, California. March 1970. (AD-705677).

72. Paz, Israel M. "Advances in Optimal Routing Through Computer Networks," NASA Technical Report R-475. September 1977.

73. Pensworth, F.L. "Review of Different Approaches to the Standard Interface Problem," Proceedings of the National Aerospace Electronics Conference, 245-252. Dayton, Ohio: Dayton Section IEEE. May 1973.

74. Porubcansky, Cynthia A. (Editor). Proceedings of the Second AFSC Avionics Standardization Conference. HQ ASD/ENAS, WPAFB, Ohio. December 1982. (ASD-ENA-TR-82-5031).

75. Reese, I.R. and D.R. Porter. "An Airplane Fiber Optic Multiplex Data Bus Design and Feasibility Demonstration System," AIAA Paper No. 77-1475. (1977).

76. Rosenkoetter, E.A. and R.C. Burns. "Advanced System Integration Demonstrations (ASID) for the 1990's," Proceedings of the National Aerospace Electronics Conference, 651-658. Dayton, Ohio: Dayton Section IEEE. May 1983.

77. Ruggles, R. and E.M. Scott. "Developments in Aircraft Digital Systems," Proceedings of the AGARD Conference: "Automation in Manned Aerospace Systems". Dayton, Ohio. October 1972.

78. Sanducci, Anthony F. "MIL-STD-1553 Bus Controller and Multiplex Remote Terminal," Proceedings of the National Aerospace Electronics Conference, 1311-1318. Dayton, Ohio: Dayton Section IEEE. May 1981.

79. Scarpino, F.A. and J.G. Weber. "A DAIS Architecture Utilizing Microprocessors," Proceedings of the National Aerospace Electronics Conference, 82-88, Dayton, Ohio: Dayton Section IEEE. May 1976.

80. Semcor Inc., Moorestown, New Jersey: A Comparison of Distributed Control and Command/Response Multiplex Systems. Technical Report under contract N62269-74-C-0761. January 1975. (AD-A025105).

81. Smith, A.L. et al. Advanced Avionic Systems for Multimission Applications. Technical Report, Boeing Military Airplane Company, Seattle, Washington. October 1982. (AD-A121793).

82. Spencer, Larry T. "Fiber Optics for the Future - Wavelength Division Multiplexing," Proceedings of the Second AFSC Avionics Standardization Conference, 871-888. HQ ASD/ENAS, WPAFB, Ohio. December 1982.

83. Sunshine, C.A. Communication Protocol Modeling. Artech House, Inc., 1981.

84. Tavora, C.J. and J.R. Glover. "The RLU - An Advanced Remote Terminal for Avionic Systems," Proceedings of the National Aerospace Electronics Conference, 1271-1278. Dayton, Ohio: Dayton Section IEEE. May 1980.

85. Treacy, J.J. "FAA Certification Criteria for Critical and Essential Digital Ssytems," SAE Technical Paper Series, No. 811060, October 1981.

86. Verdi, James S. High Speed Multiplex Bus Protocol Study. Technical Report, Naval Air Development Center, Warminster, Pennsylvania. June 1980. (AD-A102113).

87. Weber J.G. et al. "DAIS: Design and Performance," AIAA Paper No. 771499. (1977).

88. Wiegand, R. "Digital Data Transform Applications," Proceedings of the National Aerospace Electronics Conference, 703-709. Dayton, Ohio: Dayton Section IEEE. May 1980.

89. Wigle, G.B. Spare Memory and Timing Parameters in Avionics Computer System Requirements. MS Thesis, Air Force Institute of Technology, WPAFB, Dayton, Ohio. December 1977. (AD-A056521).

90. Wilson, Dayton H. "A Polled Contention Multiplex System Using the MIL-STD-1553 Protocol," Proceedings of the Fourth AIAA/IEEE Digital Avionic Systems Conference, 264-270. Dayton, Ohio: Dayton Section, IEEE, June 1981.

91. Windham, W.A. "An Application of Simulation to the Design Formulation of a Helicopter Integrated Multiplex System," Proceedings of the National Aerospace Electronics Conference, 682-687. Dayton, Ohio: Dayton Section IEEE. May 1983.

92. Wright, R.E. "Avionics System Architecture," Principles of Avionics Computer Systems (Chapter 6), December 1974. (AGARD-AG-183).

VITA

Mensur F. Krilic was born on September 3, 1942, in Sarajevo, Yugoslavia. He graduated from the First Classical High School in Sarajevo in 1961 and from the Yugoslav Air Force Technical Academy in 1964, receiving a commission in the Yugoslav Air Force. He also graduated from the Faculty of Electrical Engineering, University of Belgrade, Yugoslavia, in 1976, receiving a Bachelor of Science degree in Electrical Engineering and has been elected as an assistant-professor in the field of Aircraft Electrical and Electronic Equipment in 1981 at the same Faculty.

He entered the United States Air Force Institute of Technology in June 1982.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS:
2a. SECURITY CLASSIFICATION AUTHORITY:	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE:	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GE/EE/83D-36	5. MONITORING ORGANIZATION REPORT NUMBER(S):
6a. NAME OF PERFORMING ORGANIZATION: School of Engineering; 6b. OFFICE SYMBOL (If applicable): AFIT/ENG	7a. NAME OF MONITORING ORGANIZATION:
6c. ADDRESS (City, State and ZIP Code): Air Force Institute of Technology, Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code):
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: ; 8b. OFFICE SYMBOL (If applicable):	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:
8c. ADDRESS (City, State and ZIP Code):	10. SOURCE OF FUNDING NOS. — PROGRAM ELEMENT NO.: ; PROJECT NO.: ; TASK NO.: ; WORK UNIT NO.:
11. TITLE (Include Security Classification): See Box 19	

12. PERSONAL AUTHOR(S): Ensur F. Krilic, Lt Col, Yugoslavian Air Force

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
MS Thesis	FROM _____ TO _____	1983 December	218

16. SUPPLEMENTARY NOTATION

Approved for public release: IAW AFR 190-17
7 Feb 84
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

17. COSATI CODES		
FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number): Avionics, Digital Avionics, Computer Networks, Network Architecture

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: A STUDY OF OPTIMAL COMPUTER NETWORK ARCHITECTURE FOR DIGITAL AVIONICS SYSTEMS

Thesis Chairman: W. D. Seward, Ph.D., Major, USAF

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code): ; 22c. OFFICE SYMBOL:

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

In this paper, current theoretical considerations in available literature have been used to sort out the essential figures of merit of computer network architectures for digital avionic systems. Fourteen different approaches to the same problem of data multiplexing in avionics systems are analyzed according to the key issues. Conclusions drawn are used to define the "optimal" computer network architecture for digital avionics.

The Self-Managing Multiplex System (SMS) is conceptually designed with respect to the "optimal" characteristics, along with the discussions of some trade-offs that had to be made. The burst errors self-correcting feature of the broadcast-acknowledgments in the SMS concept seems to deserve some sort of testing in practice. It is recommended that a detailed simulation study should be performed later and a hot bench built up using the latest technologies that exist.


FILM
3-8